TK85

DOMÍNIO RÁPIDO

NÉLSON CASARI

atlas

TK 85
DOMÍNIO
RÁPIDO

EDITORA ATLAS S.A.
Rua Conselheiro Nébias, 1384 (Campos Elísios)
Caixa Postal 7186 – Tel.: (011) 221-9144 (PABX)
01203 São Paulo (SP)

NÉLSON CASARI

SÃO PAULO
EDITORA ATLAS S.A. — 1986

Rua Conselheiro Nébias, 1384 (Campos Elísios)
Caixa Postal 7186 – Tel.: (011) 221-9144 (PABX)
01203 São Paulo (SP)

ISBN 85-224-0133-0

Impresso no Brasil/**Printed in Brazil**

Depósito legal na Biblioteca Nacional, conforme Decreto nº 1.825, de 20 de dezembro de 1907.

Capa
Paulo Ferreira Leite

Ilustrações e composição
Atelier Marco Casari – São Paulo, SP (011) 247-2112

Dados de Catalogação na Publicação (CIP) Internacional
(Câmara Brasileira do Livro, SP, Brasil)

Casari, Nélson, 1931-
C33t TK 85, domínio rápido / Nelson Casari. -- São Paulo : N. Casari, 1986.

ISBN 85-224-0133-0

1. BASIC (Linguagem de programação para computadores) 2. Microcomputadores 3. Programas de computador I. Título.

17. CDD-651.8
18. -001.6404
18. -001.6424
85-1947 18. -001.6425

Índices para catálogo sistemático:

1. BASIC : Linguagem de programação : Computadores : Processamento de dados 651.8 (17.) 001.6424 (18.)
2. Microcomputadores : Processamento de dados 651.8 (17.) 001.6404 (18.)
3. Programas : Computadores : Processamento de dados 651.8 (17.) 001.6425 (18.)

- A todos aqueles que têm sede de aprender a saber e que, aprendendo, continuam humildes.

- A todos aqueles que, no anonimato, alargam as fronteiras da tecnologia eletrônica, permitindo a realização da pequena maravilha que é o microcomputador.

- A todos aqueles cujas realizações contribuíram para a popularização da computação, particularmente a Sir Sinclair Lewis.

INDICE

APRESENTAÇÃO

Quem compra um aparelho utilitário normalmente deseja fazer uso amplo e imediato do mesmo. Isto é mais válido quando se trata de um microcomputador pessoal, que, diga-se de passagem, de pessoal mesmo, estritamente, só tem o nome, pois qualquer deles, com um mínimo de 16 K "bytes" de memória RAM, encontra aplicações profissionais incontestáveis no lar ou na empresa.

E exatamente em consequência da amplidão de suas possibilidades e de sua multiplicidade de recursos, exige, ao contrário de outros utilitários, um conhecimento sempre mais profundo e atualizado da maneira de explorá-lo adequadamente, para seu domínio total e perfeito.

Seu universo de aplicações se amplia inexoravelmente. A cada dia novos recursos para sua utilização são descobertos, graças a milhares de incansáveis e apaixonados usuários e estudiosos que procuram sempre obter o máximo de rendimento dessas pequenas maravilhas.

De forma geral, o fabricante não se preocupa em ir incorporando ao manual de uso de seu produto informações adicionais sobre novos usos e recursos descobertos, de tal sorte que a maioria de tais manuais se torna desatualizada. Por outro lado, alguns são mal traduzidos ou mal redigidos, tornando-se pouco inteligíveis.

Surgiu desse fato a idéia de compilar TK 85 - DOMINIO RAPIDO, um manual que possa se tornar um auxiliar efetivo dos usuários de microcomputadores da linha SINCLAIR, em especial do TK 85 e CP 200 S.

Acreditamos estar faltando um manual que, por sua praticidade e facilidade de consulta, constitua-se numa fonte organizada de informações, dados, tabelas, programas e dicas de acesso rápido, ao lado do computador ou em cima do joelho, naqueles momentos em que surge a necessidade de um elemento chave para a execução de um programa ou de um estudo e em que não ocorre à memória a solução.

A idéia, enfim, é colocar à disposição do usuário de microcomputador elementos que o auxiliem a atingir mais rapidamente um bom grau de domínio de seu aparelho, para sua ampla utilização no espaço de tempo mais curto possível após sua aquisição.

Um ligeiro exame do seu índice permitirá sentir esse propósito.

Esperamos tê-lo atingido plenamente, para completa satisfação de seu possuidor.

OS COMANDOS - OU INSTRUÇÕES - DO TK 85

Os comandos - ou instruções - do TK 85 foram estabelecidos em linguagem BASIC, a linguagem de computação de maior proximidade com o linguajar humano, em inglês. Quando postos em execução, os comandos passam pelo dispositivo interpretador do microcomputador e acionam os circuitos eletrônicos específicos a cada ação ordenada, fazendo com que sejam desempenhadas as funções projetadas para o aparelho.

Os comandos do TK foram agrupados por nós segundo sua forma de atuação, de maneira a facilitar a compreensão de seu funcionamento e tornar sua assimilação mais rápida.

COMANDOS DE AÇÃO DIRETA

São aqueles que só podem ser acionados isolada ou independentemente, não sendo programáveis, isto é, não podem ser colocados como linhas de instrução em programas, para execução.

COMANDOS DE AÇÃO DIRETA E/OU PROGRAMAVEL

São aqueles que podem ser acionados independentemente ou atuar como instruções específicas em linhas de programas.

COMANDOS DE AÇÃO APENAS PROGRAMAVEL

São aqueles que só funcionam como instruções específicas em linhas de programas.

Uma tabela de todos os comandos do TK 85 é apresentada na página seguinte. Suas funções ou atuações são descritas sucintamente e com antecedência aos programas em que são ministrados, no capítulo BASIC SEM RODEIOS, onde são introduzidos sucessivamente, com uso exemplificado e com comentários elucidativos.

As páginas em que os comandos são introduzidos estão indicadas na tabela, de maneira a facilitar sua localização para consultas.

TABELA DOS COMANDOS DO TK 85

AÇÃO DIRETA

Comando	pg.
BREAK	23
CONT	24
EDIT (S)	29
FUNCTION (S)	33
GRAPHICS (S)	45
NEW LINE	15
SHIFT	16
RUBOUT (S)	16
⇧ ⇩ ⇦ ⇨ (S)	38

AÇÃO DIRETA E/ OU INDIRETA

Comando	pg.
CLEAR	28
CLS	24
COPY	62
FAST (S)	50
GOTO	22
LEN (F)	48
LET	26
LIST	19
LLIST (S)	62
LOAD	62
LPRINT (S)	62
NEW	15
PLOT	38
POKE	56
PRINT	15
RAND	32
RUN	15
SAVE	62
SLOW (S)	50
STOP (S)	24
UNPLOT	38

AÇÃO INDIRETA

Comando	pg.
AND (S)	38
AT (S)	18
CHR$ (F)	43
CODE (F)	36
DIM	54
FOR	26
GOSUB	24
IF	32
INKEY$ (F)	36
INPUT	32
INT (F)	32
NEXT	26
NOT (F)	35
OR (S)	36
PAUSE	24
PEEK (F)	56
REM	24
RETURN	24
RND (F)	32
SCROLL	23
STEP (S)	40
STR$ (F)	48
TAB (F)	19
THEN (S)	32
TO (S)	26
USR (F)	58
VAL (F)	53

FUNÇÕES MATEMÁTICAS

Função	pg.
ABS (F)	108
ACS (F)	113
ASN (F)	113
ATN (F)	113
COS (F)	113
EXP (F)	110
LN (F)	111
PI (F)	111
SGN (F)	108
SIN (F)	113
SQR (F)	112
TAN (F)	113

ESPECIAIS (AÇÃO INDIRETA)

Comando	pg.
" " (S)	17
, (S)	20
; (S)	20
() (S)	108

(S) = Acionáveis simultaneamente com a tecla SHIFT.

(F) = Acionáveis apenas após o comando FUNCTION ser ativado.

CODIGOS DE REPORTAGEM OU INDICADORES

São códigos que aparecem como notações no canto inferior esquerdo do vídeo, indicando situações ou estados do microcomputador face a instruções de programas ou comandos feitos.

As notações aparecem sempre na seguinte configuração: NUMERO OU LETRA DO CODIGO - BARRA - NUMERO DE LINHA.

A seguir os códigos comuns e sua significação:

0 - Execução bem sucedida.

1 - Existência de instrução NEXT sem definição de variável de controle.

2 - Utilização de variável indefinida.

3 - Utilização de variáveis subscritas fora de faixa.

4 - Memória insuficiente ou esgotada.

5 - Falta de espaço na tela de TV. O comando CONT cria espaço, limpando a tela.

6 - Sobrecarga aritmética.

7 - Comando RETURN usado sem o correspondente GOSUB.

8 - Uso indevido do comando INPUT.

9 - Comando STOP executado.

A - Argumento inválido para certas funções como SQR, LN, ASN.

B - Número inteiro fora de faixa adequada para arredondamento.

C - Expressão numérica inválida na instrução VAL.

D - Programa interrompido por BREAK.

E - Não utilizado.

F - Comando SAVE utilizado em "string" vazia.

O BASIC SEM RODEIOS, ATRAVES DO USO

Sabemos que as crianças aprendem a falar espontaneamente, sem teorias, apenas fazendo uso das palavras, cujo significado só mais tarde assimilam. E logo estão se comunicando com o mundo, sem rodeios ou teorias, naturalmente.

Vamos aprender a linguagem BASIC do microcomputador através desse método: o uso direto, sem teorias. E quando quisermos conhecê-las, vamos consultar sua gramática.

As teclas do TK englobam as seguintes funções:

Algumas teclas não são providas de todas essas funções.

As palavras referentes a comandos, instruções ou funções não são digitadas letra por letra. Exemplos: PRINT, RUN, LIST etc. Basta pressionar a tecla correspondente para acioná-las. O espaço após as mesmas também não precisa ser digitado, pois faz parte delas.

O espaço entre outras palavras ou números em digitação é dado pela tecla SPACE.

Mãos à obra! Ligue o computador e a TV.
Espere aparecer o CURSOR K no canto inferior esquerdo do vídeo.

COMANDOS PRINT, RUN E NEW LINE

PRINT: Faz aparecer no vídeo o resultado de um comando, seja um número, uma palavra ou frase, um desenho, uma operação matemática ou qualquer outra informação, inclusive espaços "vazios".

RUN: Põe em execução um programa existente no computador.

NEW LINE: Aciona os comandos introduzidos via teclado, encaixa no programa as linhas de instruções digitadas e faz aparecer listado no vídeo um programa interrompido ou já executado.

Digite em seguida as frases abaixo, sem esquecer dos números e aspas. Após digitar cada frase, pressione a tecla NEW LINE.

```
1 PRINT "COMEÇO AGORA"
2 PRINT "A FAZER USO"
3 PRINT " DO MEU TK"
```

Note que após ter sido digitado um comando o CURSOR K passa para L, indicando que está aguardando a entrada de dados.

A numeração adotada nas linhas acima tem a finalidade de caracterizar um programa, para que o computador o execute na sequência indicada. Faça "rodar" o programa:

Tecle RUN e NEW LINE. Admire a execução de seu primeiro programa no vídeo.

Tecle agora NEW LINE. Olhe novamente para o vídeo.

Tecle de novo RUN e NEW LINE.

COMANDO NEW

NEW: Limpa toda a memória de programas (RAM) do computador, zerando o sistema. NEW é a primeira tecla à esquerda da terceira carreira (de cima para baixo) do teclado. Não deve ser confundida com a tecla NEW LINE.

Limpe agora a memória do microcomputador, isto é, ponha o mesmo na condição de "livre para receber um programa":

Tecle NEW e NEW LINE.

COMANDOS SHIFT E RUBOUT

SHIFT: Comutador do teclado. Acionado com FUNCTION, permite introduzir os comandos e funções impressos em branco fora e abaixo das teclas. Acionado com GRAPHICS e as teclas correspondentes, introduz os símbolos gráficos. Acionado com as demais teclas, introduz os comandos e sinais impressos em amarelo nas mesmas.

RUBOUT: Funciona apenas em linhas em edição (digitação) ou em linhas já editadas, de programas, quando estas são levadas para o local de edição através do comando EDIT. Atua nas posições onde se posicionarem os cursores de edição (K, L, F, G), sendo estes deslocados através das teclas 5 e 8 (nas direções das setas de cada tecla), pressionadas as mesmas simultaneamente com a tecla SHIFT.

Faça agora seu segundo programa.

Digite, não esquecendo de teclar NEW LINE após cada linha:

```
1 PRINT "GOSTEI."
2 PRINT "VOU CONTINUAR."
3 PRINT "O QUE FAÇO AGORA?"
4 PRINT "FAREI CONTAS..."
```

Obs.: Se durante a digitação pressionar indevidamente alguma tecla, tecle simultaneamente SHIFT e RUBOUT em seguida para apagar o carácter introduzido assim, e prossiga normalmente com a digitação.

Vão ser introduzidos a seguir os sinais de operações aritméticas em continuidade ao programa acima iniciado:

Digite:

```
5 PRINT "2*2"
6 PRINT "10/2"
7 PRINT "10+10"
8 PRINT "20-10"
```


Note que os sinais + e - são os convencionais. No microcomputador o sinal de multiplicação é o * (asterisco) e o de divisão a /(barra).

Tecle RUN e NEW LINE. Repare no vídeo que as contas ou operações indicadas não foram realizadas. O que está entre aspas após a instrução PRINT o computador apenas "imprime" no vídeo.

Digite, agora sem aspas:

```
 9 PRINT 2*2
10 PRINT 10/2
11 PRINT 10+10
12 PRINT 20-10
```

Tecle RUN e NEW LINE. Examine o vídeo e note que as operações aritméticas das linhas 9, 10, 11 e 12 foram agora realizadas, por estarem sem aspas.

ASPAS COMO COMANDOS

As ASPAS indicam ao computador que quaisquer caracteres contidos por elas devem ser "escritos" no vídeo, podendo ser apenas um carácter qualquer, números, um desenho, uma palavra, frase ou um texto, constituindo ou caracterizando uma "string". As mesmas não são reproduzidas nas instruções executadas que as precedem.

Quando se deseja usá-las na sua função normal, isto é, como símbolos gráficos, de maneira a destacar algo (uma palavra ou frase, por exemplo), deve ser acionada a tecla que as contém duplicadas, em amarelo (tecla Q).

Tecle NEW e NEW LINE para zerar o microcomputador.

Digite:

```
10 PRINT "2*2 = ";2*2
20 PRINT "10/2 = ";10/2
30 PRINT "10+10 = ";10+10
40 PRINT "20-10 = ";20-10
```

Tecle RUN e NEW LINE. Observe no vídeo o efeito exercido pelos sinais de ponto e vírgula colocados após as aspas e seguidos pelas operações indicadas sem aspas.

Digite, sem se importar com o que já está digitado, ou seja o programa de linhas 10, 20, 30 e 40:

```
 1 PRINT "OPERAÇÕES MATEMATICAS:"
 5 PRINT "MULTIPLICAÇÃO:"
15 PRINT "DIVISÃO:"
25 PRINT "ADIÇAO:"
35 PRINT "SUBTRAÇÃO:"
```

Observe no vídeo como ficaram ordenadas as instruções da primeira parte digitada do programa - linhas 10, 20, 30 e 40 - e as da segunda parte digitada - linhas 1, 5, 15, 25 e 35. Se a primeira parte tivesse sido digitada com números de linhas de 1 a 4 não teria sido possível intercalar as linhas da segunda parte. O programa todo teria que ser digitado novamente desde a primeira linha. Veja nesse exemplo a conveniência de qualquer programa ser digitado com numeração de linhas espaçada, em especial os programas longos e complexos, para que instruções suplementares ou "esquecidas" possam ser introduzidas devidamente, isto é, no local certo e a qualquer tempo.

Tecle RUN e NEW LINE. Observe o efeito do programa e o resultado das instruções suplementares.

Tecle NEW e NEW LINE, zerando o computador.

Até agora o comando PRINT escreveu ou imprimiu no vídeo o que lhe foi instruído sempre a partir do ponto extremo do canto superior esquerdo da tela. Vamos agora usar outros pontos ou locais do vídeo.

COMAND AT

AT: Determina a posição para PRINT, indicando linha e coluna para início de impressão.

Digite:

```
 5 PRINT "."
10 PRINT AT 5,5;"AQUI LINHA 5, COLUNA 5"
20 PRINT AT 10,10;"AQUI 10,10"
30 PRINT AT 15,15;"AQUI 15,15"
40 PRINT AT 17,20;"AQUI 17,20"
50 PRINT AT 21,31;"."
```


Tecle RUN e NEW LINE. Observe bem as posições das impressões feitas no vídeo e compare-as com as instruções do programa.

Tecle NEW LINE.

Digite, acrescentando ao programa acima:

```
 7 PRINT "POS. PONTO: LINHA 0, COLUNA 0"
15 PRINT AT 21,0;"POS. PONTO: LINHA 21, COLUNA 31"
```

Tecle RUN e NEW LINE. Observe que o comando PRINT, quando não é seguido da indicação do local a imprimir, imprime na linha seguinte à da execução da instrução PRINT que a precede. E quando seguido da indicação do local a imprimir, executa à risca a ordem, não importando onde se encontre, como no exemplo da linha 15.

COMANDO LIST

LIST: Faz com que seja apresentada no vídeo a listagem do programa que estiver na memória do computador.

Tecle LIST e NEW LINE e examine mais uma vez a listagem que será exibida no vídeo, no caso a do programa que introduziu AT.

Tecle RUN e NEW LINE. Repare agora no canto inferior esquerdo do vídeo. Há ali um código de reportagem, ou uma denotação: 0/50. Nº 0 indica que o programa foi executado, isto é, que a ordem de comando RUN foi cumprida. O nº 50 após a barra indica a última linha de programa executada, nesse caso a final, significando que o programa completou todo seu ciclo, sem interrupção. Consulte a tabela completa dos códigos de reportagem para se familiarizar com as demais indicações.

Tecle NEW e NEW LINE para "zerar" o microcomputador.

COMANDO TAB

TAB: Determina a coluna para início de impressão no vídeo, sendo a largura da tela considerada em 32 colunas, numeradas de 0 a 31, partindo da margem esquerda.

Digite:

```
10 PRINT "COLUNA 0"
```

```
20 PRINT TAB 5;"COLUNA 5"
30 PRINT TAB 14;"COLUNA 14"
40 PRINT TAB 22;"COLUNA 22"
```

Tecle RUN e NEW LINE. Note o efeito de TAB no vídeo.

LINHAS E COLUNAS

Para efeito de determinar locais de impressão na tela de vídeo, devemos saber então que a mesma se divide em:

22 linhas numeradas de 0 a 21, a partir da margem superior, e

32 colunas numeradas de 0 a 31, a partir da margem esquerda.

Para determinar o local a imprimir, a instrução deve ser feita na ordem LINHA, COLUNA, através dos comandos PRINT e AT. Para indicar apenas a COLUNA como local de início de impressão, deve ser usada a instrução TAB.

Quando não houver indicação de local após o comando PRINT, a impressão será iniciada pelo computador na primeira linha que, na ordem de sequência de execução do programa, estiver disponível, e sempre na coluna 0, ou após um sinal de ponto e vírgula existente no final de uma linha de instrução PRINT precedente.

Tecle RUN e NEW LINE para zerar o micro.

VIRGULA E PONTO E VIRGULA COMO COMANDOS ESPECIAIS

Embora a rigor não sejam definidos como tais, seu desempenho em linhas de programas obriga que assim sejam considerados.

VIRGULA: Determina colunas para início de impressão no vídeo. Uma vírgula colocada após o comando PRINT determina como local de início de impressão no vídeo a coluna 16. Duas vírgulas após PRINT deslocam o início de impressão para uma segunda linha, coluna 0, deixando a primeira em branco. Três vírgulas após PRINT deslocam o início de impressão para a coluna 16 da segunda linha. Quatro vírgulas seguindo a PRINT deslocam a impressão para uma terceira linha, coluna 0, saltando duas. E assim por diante.

O seu uso com essa finalidade economiza bytes num programa

A vírgula é também usada obrigatoriamente após a instrução POKE. BYTES e POKE serão vistos mais adiante.

PONTO E VIRGULA: O conjunto convencional - ponto em cima, vírgula em baixo - determina como local para início de impressão a posição imediatamente seguinte à sua, quando colocado no final de instruções que contenham o comando PRINT. E usado obrigatoriamente com o comando conjunto PRINT AT, após o número de coluna indicado.

Digite:

```
10 PRINT "LINDO"
20 PRINT "DIA"
30 PRINT "LINDO",
40 PRINT "DIA"
50 PRINT "LINDO";
60 PRINT "DIA"
```

Tecle RUN e NEW LINE. Perceba no vídeo o efeito da vírgula colocada na linha 30 e do ponto e vírgula na linha 50.

Tecle NEW LINE. Para que a última frase do programa fique correta, acrescente um espaço entre a palavra LINDO e a aspa que a segue:

Digite novamente a linha 50:

```
50 PRINT "LINDO ";
```

Tecle NEW LINE. A substituição da linha 50 anterior pela modificada se processará automaticamente.

Tecle RUN e NEW LINE. Note a diferença na última frase depois da alteração feita e estude mais uma vez o efeito da pontuação.

LINHAS DE INSTRUÇÕES COM NUMERAÇÃO IDENTICA

Uma linha de instrução digitada com numeração já existente em outra sobrepõe-se à mesma, apagando-a. Deve-se tomar cuidado ao digitar os números de linhas de instruções de um programa. Se houver no mesmo uma linha já digitada com número 1, por exemplo, poderá suceder que se no momento de digitar a de número 10 a tecla 0 não for pressionada adequadamente e passar despercebida a falha, a que deveria ter nº 10 fica sendo mesmo linha 1 e vai se colocar no lugar da que já existia, apa-

gando-a. Essa é uma das razões que aconselham a sempre conferir com cuidado um programa após ser digitado, antes de rodá-lo.

Tecle NEW e NEW LINE para zerar o computador.

Digite: 10 PRINT "VOU PARAR";AT 5,0;"ESTOU CANSADO";
AT 20,10;"CONTINUAREI DEPOIS";AT 10,5;
"MAS ESTOU GOSTANDO"

Atente bem para a colocação das aspas, vírgulas e pontos e vírgulas.

Tecle RUN e NEW LINE. Perceba o alcance de uma única instrução com auxílio do comando AT e do sinal ponto e vírgula.

Tecle NEW e NEW LINE. Já sabemos que assim zeramos o micro.

Digite:

```
10 PRINT "NÃO"
20 PRINT "GOSTO DE ESTUDAR."
30 PRINT "VOU PASSEAR."
40 PRINT "VOU ME DIVERTIR."
```

Tecle RUN e NEW LINE. Atente bem ao que o vídeo exibe.

Digite RUN 20 e tecle NEW LINE. Leia o que está no vídeo agora.

Digite RUN 30 e tecle NEW LINE. Leia o que o vídeo exibe agora.

Digite RUN 40 e tecle NEW LINE. Observe novamente o vídeo.

COMANDO GOTO

GOTO: Dirige ou desvia a execução de um programa para a linha do mesmo cujo número é apontado por ele.

Digite GOTO 10 e tecle NEW LINE.

Digite GOTO 20 e tecle NEW LINE.

Digite GOTO 30 e tecle NEW LINE.

Observe sempre o que é escrito no vídeo após a execução de cada comando feito, seguido ou não do número de linha de programa. Compare-os bem.

Digite GOTO 40 e tecle NEW LINE.

Tecle NEW LINE.

Tecle LIST e NEW LINE.

Digite LIST 20 e tecle NEW LINE.

Digite LIST 30 e tecle NEW LINE.

Digite LIST 40 e tecle NEW LINE.

Tecle NEW e NEW LINE.

Digite:

```
10 PRINT "MEU NOME E JULIO"
20 GOTO 10
```

Tecle RUN e NEW LINE. Observe o efeito de GOTO na linha 20.

Tecle NEW LINE.

Acrescente agora uma linha ao programa:

Digite: 5 SCROLL

Mude a linha 20:

Digite: 20 GOTO 5

COMANDO SCROLL

SCROLL: Faz com que a impressão no vídeo seja feita a partir da última linha inferior, sucessivamente, empurrando para cima as linhas que vão sendo exibidas e eliminando as que vão atingindo o topo da tela de vídeo.

Tecle RUN e NEW LINE. Observe a frase "rolando" no vídeo.

COMANDO BREAK

BREAK: Detém a execução do programa em certas situações. Não

atua após o comando INPUT.

Tecle BREAK e NEW LINE, brecando o programa que está rodando.

Tecle NEW e NEW LINE.

COMANDOS GOSUB, STOP, CONT, REM, PAUSE, CLS E RETURN

GOSUB: Desvia a execução de um programa para uma subrotina constante do mesmo, retornando automaticamente a execução para a linha de instrução seguinte à do desvio.

STOP: Detém a execução de um programa nas seguintes situações:

- quando for instrução em linha de programa;
- quando acionado simultaneamente com a tecla SHIFT enquanto aguarda entrada de dados para uma variável numérica na execução de uma instrução INPUT;
- quando acionado simultaneamente com a tecla SHIFT após esta ter sido acionada previamente com uma das teclas EDIT ou RUBOUT, enquanto aguarda entrada de dados para uma variável alfanumérica na execução de uma instrução INPUT.

CONT: Reinicia a execução de um programa interrompido pelo comando STOP ou por falta de espaço no vídeo, a partir do ponto em que ocorreu a interrupção.

REM: Torna sem efeito de execução uma linha de programa, quando colocado após o número da mesma. Permite também a inclusão de linhas com comentários elucidativos em um programa, sem que o computador as interprete como comandos ou linhas de instrução do programa.

PAUSE: Detém a execução do programa pelo tempo cujo valor lhe é atribuído, mantendo no vídeo a última imagem gerada antes de sua execução.

CLS: Elimina do vídeo as imagens ou dados produzidos pelo programa.

RETURN: Comando complementar obrigatório de GOSUB, devendo figurar sempre na última linha de uma subrotina.

Vamos experimentar os comandos acima num pequeno programa:

Digite:

```
 10 PRINT AT 10,8;"ATENÇÃO, AGUARDE"
 20 GOSUB 160
 30 PRINT AT 10,9;"VEM AI UMA BOA"
 40 GOSUB 160
 50 PRINT AT 10,5;"NÃO, NÃO E ISSO, NÃO..."
 60 GOSUB 160
 70 PRINT AT 10,6;"NADA DE RACHEL WELCH"
 80 GOSUB 160
 90 PRINT AT 10,7;"SABE QUAL E A BOA?"
100 GOSUB 160
110 PRINT AT 10,4;"BEM, SO VOCE E QUEM SABE"
120 GOSUB 160
130 PRINT AT 10,8;"FIM, OBRIGADO"
140 STOP
150 REM SUBROTINA DE REPETIÇÃO
160 PRINT AT 7,12;"TK NEWS"
170 PAUSE 300
180 CLS
190 RETURN
```

Tecle RUN e NEW LINE.

Note que, para fazer um programa que a cada poucas linhas deva repetir algo - uma operação aritmética, uma tabela, um texto etc. - se não houver a instrução GOSUB, que desvia o programa para uma subrotina estabelecida para repetição, todas as instruções ou dados da mesma terão que ser inseridos tantas vezes quanto forem exigidos. No programa acima, por exemplo, no lugar das linhas 20, 40, 60, 80, 100 e 120. Se for uma subrotina pequena como a iniciada na linha 160, não será tão difícil ou trabalhoso. Mas, imagine uma subrotina de 20 ou mais linhas com instruções complexas! A instrução GOSUB elimina esse problema. Exige apenas que se coloque no fim da subrotina a instrução que lhe é complementar: RETURN.

Repare também na instrução REM da linha 150, que serve apenas de guia ou lembrança para o programador ou usuário do computador. Não aciona nenhuma função e não impede o andamento do programa.

Tecle RUN e NEW LINE, rodando novamente o programa para observar bem os efeitos dos comandos PAUSE e CLS.

Ao parar o programa, note no canto inferior esquerdo do vídeo a

denotação 9/140, significando que o programa parou pelo comando STOP da linha 140. A linha 140 não é a última linha de instrução do programa, mas é a de finalização, já que as demais, 150 a 190, constituem uma subrotina com instrução obrigatória de retorno.

Sinta melhor o efeito do comando PAUSE alterando a linha 170:

Digite: 170 PAUSE 100

Tecle NEW LINE para substituir a linha 170 anterior.

Tecle RUN e NEW LINE. Repare no tempo de duração de cada tela agora.

Elimine a seguir a linha 180:

Digite apenas o número da mesma: 180

Tecle NEW LINE para que seja efetuada a eliminação da mesma.

Tecle RUN e NEW LINE. Note a falta do comando CLS.

Tecle CLS e NEW LINE.

Tecle LIST e NEW LINE.

Tecle NEW e NEW LINE.

Digite em seguida, praticando um pouco mais:

```
10 PRINT ,"POR HOJE CHEGA"
20 PRINT ,,"SO AMANHÃ AGORA"
30 PAUSE 30
40 CLS
50 GOTO 10
```

Tecle RUN e NEW LINE. Não espere muito, se estiver cansado.

Tecle BREAK, NEW, NEW LINE. Desligue o micro. Descanse e reveja o que foi visto até aqui antes de prosseguir. Elimine dúvidas.

COMANDOS FOR, NEXT, LET E TO

FOR: Define e introduz uma variável contadora ou de controle, criando um "loop", ou seja um laço. Deve ser complementado por NEXT.

NEXT: Elemento acionador da sequência de contagem de uma variável de controle ou contadora.

LET: Permite atribuir um valor a uma variável numérica ou dados a uma variável alfanumérica, neste caso sempre entre aspas, caracterizando uma "string".

TO: Elemento operador complementar do comando FOR. Usado também em operações de fatiamento de "strings" ou expressões alfanuméricas, como será visto adiante.

Digite:

```
10 LET A=5
20 FOR B=1 TO A
30 PRINT B
40 NEXT B
```

Tecle RUN e NEW LINE. Atente para o último número impresso no vídeo e compare-o com o valor atribuído à variável A.

Tecle NEW LINE.

Digite: 10 LET A=10

Tecle NEW LINE.

Tecle RUN e NEW LINE. Observe de novo o último número impresso no vídeo e o valor atribuído à variável A.

O que foi feito? Foram usadas duas variáveis, A e B, a primeira como variável simples, à qual foram atribuídos valores através do comando LET, a segunda como variável contadora, à qual foi determinada a execução de contagem de 1 até o valor de A, através das instruções complementares FOR e NEXT, sendo a contagem registrada através do comando PRINT.

As variáveis poderiam ser quaisquer de A a X, como contadoras,- de A a X, de A1 a X1, AM, AR, AMER etc., como variáveis numéricas simples.

Tecle NEW e NEW LINE.

Digite:

```
10 LET M=5
20 FOR C=1 TO M
30 PRINT "ENTENDI"
40 NEXT C
```

Tecle RUN e NEW LINE.

Tecle NEW LINE e acrescente três linhas ao programa acima:

Digite:

```
 5 LET O=10     (letra O para a variável)
50 PRINT "MUITO BEM"
60 PRINT "MEREÇO NOTA ";O          (letra)
```

Tecle RUN e NEW LINE. Examine o programa executado.

Tecle NEW LINE. Estude o programa listado.

Tecle NEW e NEW LINE.

Digite:

```
10 LET A=5
20 LET B=10
30 PRINT "A+B = ";A+B
40 PRINT "B/A = ";B/A
50 PRINT "A*B = ";A*B
60 PRINT "B-A = ";B-A
```

Tecle RUN e NEW LINE. Note que as operações indicadas foram executadas de acordo com os valores atribuídos às variáveis A e B.

Tecle NEW LINE. Deixe o programa listado no vídeo.

COMANDO CLEAR

CLEAR: Anula todos os valores atribuídos a variáveis numéricas e dados atribuídos a alfanuméricas, "limpando" todas as variáveis do programa, sem apagar ou destruir este. A execução do comando RUN aciona automaticamente CLEAR, isto é, antes de o programa rodar, todas as variáveis são canceladas.

Vamos introduzir no programa acima o comando CLEAR.

Digite: 25 CLEAR

Tecle NEW LINE para introduzir a linha no programa.

Tecle RUN e NEW LINE.

O que aconteceu? O comando CLEAR da linha 25 anulou as atribuições de valores às variáveis A e B, feitas através das linhas 10 e 20, e a execução do programa parou por falta de elementos para prosseguir. Observe no canto inferior esquerdo do vídeo a denotação 2/30, o código de reportagem 2 antes da / significando utilização de variável ou variáveis sem definição.

Tecle NEW LINE. O CURSOR DE EDIÇÃO deve estar localizado na linha 25, a que foi digitada por último e inserida no programa já existente. Deixe o programa listado no vídeo.

COMANDO EDIT

EDIT: Faz com que uma duplicação da linha de programa em que esteja o CURSOR DE EDIÇÃO se reproduza no local onde são editadas as linhas de instruções - extremo inferior do vídeo - para alterações.

Tecle simultaneamente SHIFT e EDIT.

Note que a linha 25 continua no programa, mas foi também reproduzida no local onde são editados os comandos e linhas de instruções.

Tecle REM. A linha 25 deverá ficar assim:

```
25 REM CLEAR
```

Tecle NEW LINE. Note que a linha alterada com auxílio do comando EDIT substituiu a original.

Tecle RUN e NEW LINE. Observe que agora os valores atribuídos às variáveis A e B não foram anulados pelo comando CLEAR, que ainda está no programa, porém inatuante por força do comando REM.

Tecle NEW e NEW LINE.

Digite: 10 PRINT A+B

Tecle RUN e NEW LINE.

O que aconteceu? Nada. O computador apenas informou - olhe no canto inferior esquerdo do vídeo a denotação 2/10 - que não foram atribuídos valores às variáveis utilizadas, A e B, não sendo possível efetuar a operação indicada na linha de instrução de número 10.

Digite agora, sem numerar a linha:

LET A=10

Tecle NEW LINE.

Digite ainda, sem numerar a linha:

LET B=50

Tecle NEW LINE.

Tecle GOTO 10 e NEW LINE. Não tecle RUN para não anular as definições de A e B. Observe no vídeo que o programa de linha única agora foi executado.

Digite PRINT A+B

Tecle NEW LINE. Observe de novo o vídeo.

Repita as operações com PRINT B/A, PRINT B-A.

Tecle RUN e NEW LINE. Observe que, agora, o programa não foi executado e apareceu de novo no canto inferior esquerdo do vídeo a denotação 2/10, porque a instrução RUN, ao ser executada, procedeu antes a um CLEAR nos registros das variáveis.

Um programa longo ou complexo que utiliza variáveis definidas, isto é, com valores atribuídos, e que eventualmente sofre uma parada ocasional, deverá receber o comando GOTO para ser posto de novo em execução, pois tal comando não anula os registros já feitos. GOTO, todavia, precisa ser complementado pelo número de linha a que deve se

dirigir, como já vimos. Assim, para rodar o programa desde o início, basta comandar GOTO 0 ou GOTO 1 e NEW LINE, não importando se a primeira linha do programa foi 1, 10 ou 200. O computador se encarrega de procurar a primeira linha existente após 0 para iniciar a execução do programa. Tal cuidado deverá ser observado principalmente com programas carregados de fitas K-7 para o computador e que utilizam variáveis, para evitar surpresas.

Digite:

```
10 LET A$="ESTOU "
20 LET B$="GOSTANDO "
30 PRINT A$+B$
```

Tecle RUN e NEW LINE. Observe o resultado.

Tecle NEW LINE.

Digite, completando o programa:

```
40 PRINT "DE ESTUDAR "
50 LET C$="MUITO "
60 PRINT A$+B$+C$
```

Tecle RUN e NEW LINE.	Observe sempre no vídeo o resul-
Tecle GOTO 40 e NEW LINE.	tado de cada comando feito, ana-
Tecle GOTO 60 e NEW LINE.	lisando-os comparativamente. Al-
Tecle CLEAR e NEW LINE.	gumas diferenças são, muitas ve-
Tecle GOTO 40 e NEW LINE.	zes, tão sutis, que podem passar
Tecle GOTO 60 e NEW LINE.	despercebidas. Se não notá-las à
Tecle RUN e NEW LINE.	primeira vista, insista um pouco.

Repita algumas vezes os passos acima e estude bem os efeitos de LET e CLEAR em relação a RUN e GOTO.

Observe também que as variáveis, quando seguidas de $, aceitam quaisquer atribuições: palavras e/ou números e/ou frases, desenhos ou símbolos gráficos. São as VARIAVEIS ALFANUMERICAS e são introduzidas

e processadas pelo computador sempre dentro de aspas.

Tecle NEW e NEW LINE.

Experimente introduzir no computador uma linha de instrução como a que segue:

Digite: 10 LET A$=ESTOU

Tecle NEW LINE. Tecle outra vez.

A linha não entra, isto é, não é aceita pelo computador e volta com uma notação S - em vídeo inverso - indicando erro de sintaxe. Faça nova experiência:

Digite: 10 LET A=10 A

Tecle NEW LINE. Esta linha também não é aceita pelo computador, voltando com a notação de erro de sintaxe. No primeiro exemplo a causa é a falta de aspas na palavra ESTOU, por ser definição de uma variável alfanumérica, no segundo pela atribuição indevida de valor, ou melhor, atribuição errada de valor à variável numérica A.

COMANDOS IF, THEN, INPUT, INT, RAND, RND

IF: Elemento introdutor de proposições condicionais cuja dedução lógica deverá ser feita pelo computador. IF tem como complemento obrigatório THEN.

THEN: Elemento operador lógico complementar de IF, introdutor e antecessor obrigatório de um comando ou uma instrução a ser programada em função da dedução feita pelo computador sobre uma proposição condicional. (IF = Se. THEN = Então.)

INPUT: Introduz variáveis num programa, às quais deverão ser atribuídos valores ou dados à medida em que o programa é executado, sendo que cada linha de programa em que conste INPUT interrompe o andamento do mesmo para esse fim, aparecendo no vídeo, no canto inferior esquerdo, o CURSOR L, sem aspas para uma variável numérica e com aspas para uma variável alfanumérica.

INT: Arredonda um número processado pelo computador, desprezando a parte decimal.

RAND: Controla a "aleatoriedade" da função RND. Se seguido de número diferente de 0 (entre 1 e 65535) fará com que a "aleatoriedade", em caso de repetição na execução do programa, seja sempre a partir de um mesmo número "sorteado".

RND: Fornece um número aleatório entre 0 e 0.99999999. A fim de que forneça números no sistema mais aleatório possível, deve ser usado sem seu "controlador" RAND.

Não será demais lembrar nesta oportunidade que o ponto substitui a vírgula em numerações no computador, em consequência de ser este originário de países onde é o sistema adotado.

COMANDO FUNCTION

FUNCTION: Permite introduzir num programa comandos e funções ou instruções que estão impressos em branco abaixo das teclas, no painel. Exemplos: INT, RND, CODE, INKEY$, SIN, COS etc. Para tanto, faz-se necessário pressioná-lo antes, e simultaneamente, com a tecla comutadora SHIFT.

Vamos aplicar em seguida, num pequeno programa, os comandos ou instruções acima introduzidos. Antes, porém, convém lembrar que os comandos e sinais impressos em amarelo na parte superior das teclas são introduzidos com auxílio da tecla SHIFT, pressionada simultaneamente.

Digite:

```
  1 REM JOGO DE ADIVINHAÇÃO
 10 LET A= INT (RND*5)
 20 IF A<1 THEN LET A=1
 30 LET A$="VOCE ACERTOU"
 40 LET B$="VOCE ERROU"
 50 PRINT AT 8,3;"DIGITE UM NUMERO DE 1 A 5"
 60 INPUT B
 70 PRINT AT 12,15;B
 80 IF B=A THEN PRINT AT 15,10;A$
 90 IF B<>A THEN PRINT AT 15,11;B$
100 PAUSE 100
110 CLS
```

```
120 CLEAR
130 GOTO 10
```

Tecle RUN e NEW LINE. Note no vídeo a frase da linha 50 e o CURSOR L, no canto inferior esquerdo, indicando que o computador está aguardando a entrada de um número ou valor, em obediência à instrução da linha 60. Essa instrução introduz a variável B, que deverá ser definida, isto é, deverá receber um valor numérico, através de digitação, pelo teclado.

Digite então qualquer número entre 1 e 5.

Tecle NEW LINE. O computador continuará a pedir números e informando se você acertou ou errou.

Tecle SHIFT e STOP simultaneamente, se o computador estiver aguardando entrada de um dado, ou BREAK se estiver na execução da linha 100 - PAUSE - e em seguida NEW LINE, para parar o programa e fazê-lo aparecer listado no vídeo. Estude atentamente suas instruções.

Note a instrução da linha 10. Ela faz com que o computador escolha aleatoriamente um número inteiro até 5 e o atribua à variável A. Mas como 0 também pode ser sorteado pelo computador e o programa determina "de 1 a 5", a linha 20 impede que seja menor do que 1 o valor a ser dado à variável A.

Repare bem o desempenho de IF e THEN com as funções = (igual), < (menor), < >(diferente), nas instruções das linhas 20, 80 e 90.

Lembre-se finalmente que há duas maneiras de definir variáveis ou atribuir valores às mesmas: uma direta e imediata, exemplo LET A=, outra através do comando INPUT, que solicita a definição apenas no momento em que é este acionado pela execução sequencial do programa, o que propicia interessantes recursos de programação, como será visto mais adiante.

O mesmo acontece com as variáveis alfanuméricas, apenas com diferença nas atribuições, já que estas, em vez de números ou valores, são definidas através de "strings" que, como já vimos, aceitam letras ou palavras, frases ou caracteres gráficos, e mesmo números, embora eles não sejam tratados normalmente como valores, a não ser através

do comando VAL, que será visto mais adiante. As definições das variáveis alfanuméricas são feitas sempre entre aspas, não sendo reconhecidas pelo computador em outra forma.

Rode o programa mais algumas vezes, para observação. Depois:

Tecle NEW e NEW LINE.

COMANDO NOT

NOT: E o elemento operador lógico (= não), complementar de IF.

Digite:

```
 10 LET A=0
 20 PRINT "POSSO LHE FAZER UMA PERGUNTA?"
 30 PRINT ,,"(SE ""SIM"" DIGITE ""SIM"" E NEW LINE.
    EXPERIMENTE DEPOIS ""NÃO"".)"
 40 INPUT A$
 50 IF NOT A$="SIM" THEN GOTO 200
 60 PRINT ,,"COM QUANTOS PAUS SE FAZ 1 CANOA?"
 70 PRINT ,,"DIGITE O NUM. E TECLE NEW LINE"
 80 INPUT PAUS
 90 IF NOT PAUS=1 THEN GOTO 120
100 PRINT ,,"ACERTOU, PARABENS"
110 GOTO 210
120 PRINT ,,"NÃO. TENTE DE NOVO",,,"(CHANCE NUMERO "
    ;A+1;")"
130 LET A=A+1
140 IF A=5 THEN GOTO 180
150 PAUSE 100
160 CLS
170 GOTO 60
180 PRINT ,,"ACABARAM SUAS 5 CHANCES. TCHAU."
190 GOTO 210
200 PRINT ,,"OK, NÃO FAREI. SINTO MUITO."
210 PAUSE 100
220 CLS
230 LIST
```

Tecle RUN e NEW LINE. Responda às perguntas do programa. Responda acertada e não acertadamente, com o propósito de verificar os efeitos das instruções do mesmo.

A instrução da linha 230, executada sempre ao fim de cada ciclo do programa, oferece a lista do mesmo no vídeo para observações ou alterações eventuais. Após efetuá-las:

Tecle RUN e NEW LINE novamente.

Repare nas linhas 10, 130 e 140. A linha 10 coloca uma variável em uso e lhe atribui o valor 0. A linha 130 lhe atribui a função de contadora, isto é, toda vez que o computador executa sua instrução - o que é feito somente se a resposta à pergunta da linha 60 for considerada errada pela execução da linha 90 - a mesma sofre acréscimo de 1. A linha 140 verifica se a variável já contou as 5 chances para resposta errada e, em caso positivo, desvia a execução do programa para a linha 180. Verifique o que esta linha e a seguinte fazem.

Note também, na linha 80, outra maneira de denominar uma variável numérica. Em vez de usar apenas uma letra, A, P, S ou outra, foi-lhe dada a identificação de PAUS. Essa permissividade do computador é muito útil em programas com uso de muitas variáveis, quando uma denominação específica a cada uso facilita a elaboração das instruções e sua posterior verificação ou alteração.

Repare agora na instrução da linha 90. Ela poderia ter sido estabelecida de duas outras maneiras: IF PAUS<>1 THEN GOTO 120 ou IF PAUS>1 THEN GOTO 120. A linha 50 poderia também ser: IF A$<>"SIM" THEN GOTO 200.

Estude bem as instruções de desvio nas linhas 50, 90, 110, 140, 170 e 190.

COMANDOS INKEY$, CODE E OR

INKEY$: Verifica se alguma tecla é pressionada durante a execução de um programa e, em caso positivo, aciona o comando correspondente à mesma ou introduz o valor, ou o carácter, ou o código da mesma.

CODE: Fornece o código de um carácter do computador ou do carácter situado em primeiro lugar numa "string", código esse representado por um número entre 0 e 255, de acordo com a tabela de caracteres do computador.

OR: Elemento operador lógico de alternativa (= ou), complementar de IF.

Digite:

```
 10 PRINT ,,"ADIVINHE QUAL E A LETRA"
 20 PRINT ,,"(PRESSIONE UMA TECLA)"
 30 IF INKEY$="" THEN GOTO 30
 40 IF CODE INKEY$=44 OR CODE INKEY$=52 OR CODE
    INKEY$=58 THEN GOTO 70
 50 IF NOT CODE INKEY$=44 OR NOT CODE INKEY$=52
    OR NOT CODE INKEY$=58 THEN PRINT ,,"TENTE
    DE NOVO"
 60 GOTO 80
 70 PRINT ,,"ACERTOU, PARABENS"
 80 FOR A=1 TO 60
 90 NEXT A
100 CLS
110 GOTO 10
```

Note que as aspas da instrução da linha 30 não são as duplas usadas como símbolos gráficos, situadas na tecla Q. São as usadas normalmente em "strings" e obtidas através da tecla P.

Tecle RUN e NEW LINE. Tente adivinhar a letra.

Tecle BREAK para parar o programa.

Repare bem no comando INKEY$. Durante o andamento do programa sua função é verificar se qualquer tecla é pressionada. Em caso positivo, informa o centro de processamento do computador para as devidas ações. Assim, a instrução da linha 30 IF INKEY$="" THEN GOTO 30 significa mais ou menos: "se não for pressionada nenhuma tecla, fique esperando", de tal maneira que o programa não prossegue enquanto isto não for feito.

A instrução da linha 40 é interpretada pelo computador como:"se o código da tecla pressionada for igual a 44, ou 52, ou 58, então imprima no vídeo ACERTOU". A linha 50 é fácil de ser compreendida em relação à linha 40.

As linhas 80 e 90 substituem a instrução PAUSE com alguma vantagem. PAUSE, ao ser executada, faz "tremer" o vídeo, enquanto que FOR e NEXT processam a mesma operação de espera suavemente. Experimente a

substituição das linhas 80 e 90:

Digite: 80 PAUSE 60

Tecle NEW LINE.

Digite: 90

Tecle NEW LINE.

Tecle RUN e NEW LINE. Observe agora o efeito de PAUSE, depois torne a introduzir as linhas 80 e 90 originais e compare.

Tecle NEW e NEW LINE.

Digite:
```
10 FOR A=1 TO 10
20 NEXT A
30 PRINT "-";
40 PRINT INKEY$;
50 GOTO 10
```

Tecle RUN e NEW LINE. Pressione sucessivamente diferentes teclas e note bem como funciona INKEY$.

Elimine as linhas 10 e 20, rode novamente o programa, pressione algumas teclas e note como aumenta a velocidade de execução do programa. Após o mesmo completar a tela, note a denotação 5/30 ou 5/40 que aparece no canto inferior esquerdo do vídeo, significando que não há mais espaço no mesmo para continuação do programa.

Tecle NEW e NEW LINE.

COMANDOS PLOT, UNPLOT, AND, ⇧ ⇩ ⇦ ⇨

PLOT: Imprime nos locais determinados por instruções um "pixel" (abreviatura de "picture element"), pequeno quadrado correspondente a 1/4 do tamanho de um carácter gráfico normal. Os locais determinados são apontados pelo encontro de coordenadas - 43 no sentido horizontal - e abcissas - 63 no sentido vertical - numeradas a partir de 0 e iniciadas no canto inferior esquerdo da tela.

UNPLOT: "Limpa" ou "apaga" as impressões feitas por PLOT, sendo sua ação dirigida através das mesmas coordenadas e abcissas usadas para seu comando oposto.

⇧ ⇩ ⇦ ⇨: Os comandos caracterizados por setas direcionais - localizados nas teclas 5, 6, 7 e 8 - movem os diversos cursores de edição, K, L, F e G, e o de programa ou localização de linhas de instruções no mesmo. Quando conjugados com o comando INKEY$ e instruções adequadas, movimentam no vídeo, nas quatro direções, "strings" na forma de desenhos ou objetos, prestando-se especialmente para a animação de jogos.

AND: Elemento operador de ligação usado para complementar a definição de uma variável numérica ou o comando condicionante IF.

Digite:
```
10 LET A=30
20 LET B=20
30 PLOT A,B
40 LET A=A-(INKEY$="5" AND A>1)+(INKEY$="8" AND
   A<60)
50 LET B=B-(INKEY$="6" AND B>1)+(INKEY$="7" AND
   B<40)
60 GOTO 30
```

Tecle RUN e NEW LINE. Observe novamente a ação de INKEY$, agora conjugada com as funções das setas direcionais das teclas 5, 6, 7, 8. Pressione uma a uma com um só toque. Depois experimente manter cada uma pressionada durante algum tempo. Faça desenhos na tela.

Note que os valores das variáveis A e B são alterados pelo comando INKEY$ das linhas 40 e 50, de forma a que os pontos a receberem o PLOT sejam indicados apenas pelo acionamento das teclas direcionais ou de movimentação.

Tecle BREAK e NEW LINE.

Digite: 35 UNPLOT A,B

Tecle RUN e NEW LINE e repita os movimentos das teclas 5, 6, 7, 8. E ainda possível desenhar na tela?

Tecle BREAK e NEW LINE. Altere um pouco o programa:

Digite:
```
40 IF INKEY$="5" THEN LET A=A-1 AND A>1
42 IF INKEY$="8" THEN LET A=A+1 AND A<60
44 IF A>60 THEN LET A=60
```

Tecle RUN e NEW LINE. Note que o programa continua funcionando como antes. As linhas 40, 42 e 44 apenas desdobraram as instruções da linha 40 original, ocupando mais "bytes" no computador. O desdobramento feito, todavia, permite melhor compreensão do mecanismo de funcionamento das instruções e do próprio programa. Compare e estude as duas formas apresentadas.

Se quiser "treinar" um pouco de desenho no computador, coloque um REM na linha 35, desenhe à vontade, em seguida elimine o programa da memória do computador com os comandos diretos:

Tecle BREAK, NEW e NEW LINE.

COMANDO STEP

STEP: Determina o "passo" de contagem de uma variável contadora ou de controle.

Digite:
```
10 CLS
20 PRINT ,,"DIGITE UM NUMERO E NEW LINE"
30 INPUT N
40 PRINT ,,"TABUADA DE: ";N; " > "
50 LET M=N*10
60 LET B=0
70 PRINT
80 FOR A=N TO M STEP N
90 LET B=B+1
100 PRINT TAB 12;N;" * ";B;" = ";A
110 NEXT A
120 PRINT ,,"MAIS UM NUMERO? (S=SIM / N=NÃO)"
130 IF INKEY$="" THEN GOTO 130
140 IF INKEY$<>"S" THEN GOTO 160
150 GOTO 10
160 CLS
170 LIST
```


Tecle RUN e NEW LINE. Note o efeito da função STEP introduzida na instrução da linha 80. A variável contadora A é "informada" que deve contar de N (que recebe um valor do teclado através da linha 30) até M (que, pela instrução da linha 50, deve ser igual a N*10), dando passos (STEP) de tantas unidades quanto for o valor de N.

Rode o programa algumas vezes para observar bem o efeito do comando STEP e das linhas 90 e 100.

Tecle NEW e NEW LINE.

Digite:
```
10 FOR A=0 TO 100 STEP 5
20 PRINT A;
30 PRINT "-";
40 NEXT A
```

Tecle RUN e NEW LINE. Observe o efeito do programa. Depois altere o valor de STEP da linha 10.

Tecle NEW LINE e observe a localização do CURSOR de programa. Deve estar na linha 40, que foi a última digitada.

Tecle simultaneamente SHIFT e ↑(tecla 7). Repita até o CURSOR se localizar na linha 10.

Tecle simultaneamente SHIFT e EDIT. A linha 10 será reproduzida no local de edição de programas ou instruções.

Tecle simultaneamente SHIFT e →(tecla 8), repetindo o movimento até o CURSOR se localizar após o número 5.

Tecle simultaneamente SHIFT e RUBOUT, o que "apagará" o número 5.

Digite 10. A linha no local de edição deverá ficar assim: 10 FOR A=0 TO 100 STEP 10.

Tecle NEW LINE para "mandar" a linha alterada para seu lugar no programa.

Antes de rodar de novo o programa, lembre-se: use o processo acima quando quiser alterar ou corrigir alguma linha de programa. Caso

o programa seja muito extenso e o cursor se achar muito afastado, tecle LIST, mais o número da linha desejada e NEW LINE. O programa aparecerá listado no vídeo a partir dessa linha. Bastará então pressionar simultaneamente as teclas SHIFT e EDIT a fim de que a mesma seja reproduzida no local de edição, para alteração ou correção. Rode em seguida o programa:

Tecle RUN e NEW LINE. Note qual foi o passo de contagem da variável A depois da alteração da linha 10. Experimente outros passos, alterando o limite da variável A para 500, por exemplo.

Tecle NEW e NEW LINE.

Digite:
```
 10 FOR A=20 TO 0 STEP -1
 20 PRINT AT A,A;"SUBIDA"
 30 NEXT A
 40 CLS
 50 FOR A=0 TO 20 STEP 1
 60 PRINT AT A,A;"DESCIDA"
 70 NEXT A
 80 CLS
 90 FOR A=20 TO 0 STEP -1
100 PRINT AT 20-A,A+5;'DESCIDA"
110 NEXT A
120 CLS
130 FOR A=0 TO 20 STEP 1
140 PRINT AT 20-A,A+5;"SUBIDA"
150 NEXT A
160 CLS
170 GOTO 10
```

Tecle RUN e NEW LINE. Deixe o programa rodar um pouco. Analise os efeitos das diferentes instruções de contagem ascendente e descendente. Note as diferenças entre as linhas 20 e 140, 60 e 100.

Tecle BREAK.

Tecle CONT e NEW LINE.

Tecle STOP e note que este comando não tem nenhum efeito em programas em movimento como o acima, isto é, "rodando".

Tecle BREAK e NEW LINE. Estude as instruções para PRINT das linhas 20, 60, 100 e 140. Compare-as em função das atribuições feitas à variável A que, como contadora, vai se alterando, ocasionando modificações nos locais onde deve se dar o PRINT.

Tecle NEW e NEW LINE.

COMANDO CHR$

CHR$: Fornece o símbolo ou carácter representado por um código do computador, de número entre 0 e 255.

Digite:
```
 10 PRINT "DIGITE UM NUMERO ENTRE 0 E 255"
 20 PRINT ,,"  (EM SEGUIDA TECLE NEW LINE)"
 30 INPUT N
 40 PRINT ,,"O NUMERO ESCOLHIDO, ";N
 50 PRINT ,,"E O CODIGO DO CARACTER: ";CHR$ N
 60 PRINT ,,"TODOS OS CARACTERES OU SIMBOLOS DO
    TK TEM UM NUMERO DE CODIGO"
 70 PRINT ,,"PARA CONTINUAR APERTE UMA TECLA"
 80 PAUSE 4E4
 90 CLS
100 GOTO 10
```

Tecle RUN e NEW LINE. Use o programa para conhecer os caracteres e os símbolos ou palavras dos próprios comandos ou funções do TK através do comando CHR$ da linha 50. Observe também a função da linha 80. Após cada rodada do programa ela faz com que o resultado ou efeito do programa permaneça no vídeo para exame até que qualquer tecla seja pressionada, quando então o programa passa para outra rodada.

Tecle simultaneamente SHIFT e STOP quando o CURSOR L estiver no canto inferior esquerdo do vídeo, indicando que o computador aguarda a entrada - via digitação - de um número para a variável N da linha 30.

Tecle NEW LINE e NEW LINE (2 vezes), a fim de que o programa seja listado no vídeo. Estude-o mais uma vez.

Com relação à lista dos símbolos dos comandos e dos caracteres do TK, cabe observar aqui que os números 67 a 111, 122 a 125 e 195 não são usados. Alguns outros, por exemplo 112 a 121, 126 e 127 referem-se

a certos comandos e funções dos quais o computador fornece apenas um sinal de interrogação quando lhe são solicitados os símbolos.

Tecle NEW e NEW LINE.

Para ver a lista completa dos códigos referidos:

Digite:

```
10 FOR A=0 TO 255
20 SCROLL
30 PRINT A;" = ";CHR$ A
40 NEXT A
```

Tecle NEW e NEW LINE quando cessar a exibição dos códigos, caso não queira vê-los novamente.

Digite:

```
10 PRINT ,," PRESSIONE UMA TECLA QUALQUER"
20 PRINT ,,"(LEMBRE-SE: NEW LINE EM SEGUIDA)"
30 INPUT A$
40 PRINT ,,"A TECLA PRESSIONADA, ";A$;", "
50 PRINT ,,"CORRESPONDE AO CODIGO ";CODE A$;" DO TK"
60 PRINT ,,"PARA CONTINUAR APERTE UMA TECLA"
70 PAUSE 4E4
80 CLS
90 GOTO 10
```

Tecle RUN e NEW LINE. Estude a instrução da linha 50. Deduza a função exercida pelo comando CODE.

Note que a instrução da linha 30 faz o CURSOR L aparecer entre aspas no canto inferior esquerdo do vídeo, pois a instrução INPUT A$ introduz uma variável alfanumérica, que é tratada como "string", como já foi visto. Observe também as instruções da linha 50, estabelecidas de maneira a que o número de código do símbolo seja impresso entre as duas partes da frase.

Convém ainda lembrar que para interromper um programa que aguarda da definição para uma variável alfanumérica os seguintes passos são necessários:

- teclar simultaneamente SHIFT e EDIT;

- teclar simultaneamente SHIFT e STOP;
- teclar duas vezes seguidas NEW LINE para ter o programa listado no vídeo; ou:
- teclar NEW e NEW LINE para zerar o computador.

COMANDO GRAPHICS

GRAPHICS: Permite que os caracteres enquadrados nesta modalidade sejam introduzidos e inverte o modo de impressão de outros caracteres para branco com fundo preto, isto é, em vídeo inverso.

Lembre-se que o comando GRAPHICS deve ser acionado simultaneamente com a tecla SHIFT para que os caracteres gráficos possam ser introduzidos, e que as teclas destes devem ser acionadas juntamente com a aludida tecla SHIFT. No modo gráfico os demais caracteres são introduzidos normalmente em vídeo inverso. A reversão ao modo normal é operada pressionando-se novamente a tecla GRAPHICS simultaneamente com SHIFT.

Digite:

```
 10 FOR C=1 TO 176
 20 PRINT "    "            (4 espaços em vídeo inverso)
 30 NEXT C
 40 LET A=1
 50 LET B=2+INT (RND*17)
 60 LET C=2+INT (RND*28)
 70 LET D=2+INT (RND*17)
 80 LET E=2+INT (RND*28)
 90 PRINT AT B,C;"*"              (* em vídeo inverso)
100 FOR I=1 TO 30
110 NEXT I
120 PRINT AT B,C;" "        (1 espaço em vídeo inverso)
130 PRINT AT D,E;"*"              (* em vídeo inverso)
140 LET A=A+1
150 IF A=30 THEN STOP
160 GOTO 50
```

Tecle RUN e NEW LINE e ponha estrelas no céu do vídeo, é claro! Observe o efeito das linhas 10 e 20, que poderiam ser:

```
10 FOR C=1 TO 704
20 PRINT " ";               (1 espaço em vídeo inverso)
ou:
```

```
10 FOR C=1 TO 88
20 PRINT "        ";    (8 espaços em vídeo inverso)
```

Em tempo: Os espaços em vídeo inverso são obtidos acionando-se o comando GRAPHICS e em seguida a tecla SPACE.

As alternativas propostas para as linhas 10 e 20 baseiam-se no fato de que para "encher" o vídeo são necessários 704 caracteres, que podem ser obtidos de várias maneiras, das quais três são as expostas.

As linhas 40, 140 e 150 controlam os ciclos das instruções PRINT que colocam estrelas no céu. As linhas 50 a 80 sorteiam os locais para surgimento das estrelas, que são colocadas no céu pelas linhas 90 e 130. A linha 120 retira a estrela colocada pela linha 90, ou, melhor, apaga-a, causando a sensação de lampejos.

Rode o programa ainda algumas vezes e note como mudam os locais de surgimento das estrelas, em função das linhas 50 a 80.

Tecle NEW e NEW LINE ao terminar um ciclo do programa.

Produza agora um letreiro luminoso de múltiplos efeitos, usando os caracteres gráficos do TK, realmente interessantes:

Digite:

```
 10 FOR C=6 TO 25
 20 PRINT AT 9,C;"*"              (* em vídeo inverso)
 30 PRINT AT 11,C;"*"             (* em vídeo inverso)
 40 NEXT C
 50 PRINT AT 10,6;"*";AT 10,25;"*"   (* em vídeo
 60 PRINT AT 10,7;" BASES E BASIC TK "   inverso)
 70 FOR C=6 TO 25 STEP 2
 80 PRINT AT 9,C;"*"
 90 PRINT AT 9,C;"*"              (* em vídeo inverso)
100 NEXT C
110 PRINT AT 10,25;"*"
120 PRINT AT 10,25;"*"            (* em vídeo inverso)
130 PRINT AT 10,7;" BASES E BASIC TK "  (em vi-
140 FOR C=25 TO 6 STEP -2               deo inverso)
150 PRINT AT 11,C;"*"
160 PRINT AT 11,C;"*"             (* em vídeo inverso)
170 NEXT C
```



```
180 PRINT AT 10,6;"*"
190 PRINT AT 10,6;"*"              (* em vídeo inverso)
200 FOR C=1 TO 6
210 PRINT AT 10,7;"  NELSON  CASARI  "
220 PRINT AT 10,7;"  NELSON  CASARI  "     (em vídeo
230 NEXT C                                  inverso)
240 GOTO 60
```

Tecle RUN e NEW LINE e estude bem o desempenho do programa. Depois veja em conjunto todos os caracteres gráficos e de vídeo inverso do TK com o seguinte programa:

Digite:

```
 10 LET L=0
 20 LET T=0
 30 FOR A=1 TO 10
 40 PRINT AT L,T;CHR$ A;"=";"C ";A
 50 LET L=L+1
 60 NEXT A
 70 FOR A=128 TO 191
 80 PRINT AT L,T;CHR$ A;"=";"C ";A
 90 LET L=L+1
100 IF L=19 THEN LET L=0
110 IF PEEK 16442=5 THEN LET T=T+8
120 NEXT A
130 PRINT TAB 24;"C=CODIGO"
140 POKE 16418,0
150 PRINT
160 PRINT ,,"AI ESTÃO OS CARACTERES GRAFICOS E DE
    VIDEO INVERSO DO TK E SEUS RESPECTIVOS CODIGOS.
    ELES SERÃO"
170 PRINT TAB 7;"MUITO UTEIS A VOCE LOGO."
```

Tecle RUN e NEW LINE. Visualize bem todos os interessantes recursos gráficos do TK. Eles serão realmente muito úteis na elaboração de desenhos, jogos animados e textos sobre os quais se queira destaque especial, inclusive títulos.

Para tê-los exibidos no vídeo não haveria necessidade de um programa elaborado como o acima. Porém, o mesmo foi necessário para ter todo o conjunto de tais caracteres, mais seus códigos respectivos, impressos de uma só vez no vídeo, e com a estética apresentada, a fim de

propiciar uma visão de conjunto dos mesmos, e comparativa.

A elaboração do programa nessa forma, outrossim, teve por finalidade adicional demonstrar alguns recursos pouco usuais para se imprimir duas sequências diferentes de números e dados em quatro segmentos subsequentes, utilizando todas as colunas (32) e linhas (24) passíveis de serem geradas pelo microcomputador no vídeo, sendo que as duas últimas linhas só se tornam disponíveis pela manipulação da variável DFSZ, do sistema, através da instrução da linha 140 do programa.

Outra variável do sistema utilizada, LINPR, linha 110 do programa, permite estabelecer o ponto em que a impressão dos dados deve ser interrompida na parte inferior de uma coluna para ser reiniciada na coluna subsequente, na parte superior.

As instruções das linhas 10, 20, 50, 90 e 100 completam o conjunto de instruções necessárias para obtenção da disposição descrita, útil para muitas finalidades em computação de dados, razão por que aconselhamos estudá-las adequadamente, para futuras aplicações

As variáveis do sistema, acima aludidas, serão vistas no capítulo ORGANIZAÇÃO DE MEMORIA DO TK 85.

COMANDOS LEN E STR$

LEN: Fornece o comprimento de uma expressão alfanumérica, ou de uma "string", em número ou quantidade de caracteres.

STR$: Permite que um número ou uma expressão numérica sejam tratados pelo computador como "string" ou expressão alfanumérica.

Digite:

```
 10 PRINT ,,"DIGITE 1 PALAVRA, TECLE NEW LINE"
 20 INPUT A$
 30 PRINT ,,A$
 40 PRINT ,,"QTDE. DE LETRAS DA MESMA: ";LEN A$
 50 PRINT ,,"DIGITE 1 NUMERO, TECLE NEW LINE"
 60 INPUT A
 70 PRINT ,,A
 80 PRINT ,,"QTDE. DE DIGITOS DO NUM.: ";LEN STR$ A
 90 PRINT ,,"PARA CONTINUAR APERTE UMA TECLA"
100 PAUSE 4E4
```



```
110 CLS
120 GOTO 10
```

Tecle RUN e NEW LINE. Execute o programa algumas vezes e note o desempenho do comando LEN nas instruções das linhas 40 e 80. Após, tente também digitar uma frase em vez de uma palavra. LEN mede diretamente o comprimento de "strings". Porém, para medir uma expressão numérica deve ser complementado por STR$, que faz com que a expressão seja considerada como "string" para efeito de ser medida.

A maneira fácil de interromper um programa como o acima é quando ele se encontra em PAUSE:

Tecle BREAK e em seguida NEW e NEW LINE.

Digite:

```
 10 PRINT ,,"DIGITE UMA PALAVRA OU UMA FRASE"
 20 PRINT ,,"(DEPOIS TECLE NEW LINE, CERTO?)"
 30 INPUT A$
 40 LET B$=""
 50 FOR C=1 TO LEN A$
 60 LET B$=B$+"="
 70 NEXT C
 80 PRINT ,,A$
 90 PRINT B$
100 PRINT ,,"LETRAS E ESPAÇOS UTILIZADOS: ";LEN A$
110 PRINT ,,"PARA CONTINUAR APERTE UMA TECLA"
120 PAUSE 4E4
130 CLS
140 GOTO 10
```

Tecle RUN e NEW LINE. Veja agora a ação de LEN aplicada na variável contadora da linha 50, depois de "medir" a palavra ou frase atribuída à variável A$. Note também, na linha 40, a atribuição feita à variável B$: ="", que equivale a uma "string" vazia, podendo ser definida depois, o que permite a execução da linha 60, que cria para a palavra ou frase entrada um sublinhado de tamanho equivalente e que é mostrado pela execução da linha 90.

Na linha 100 de novo LEN, medindo as letras e os espaços utilizados por A$ e fornecendo o resultado.

Depois de rodar o programa algumas vezes:

Tecle BREAK.

Tecle LIST 60 e NEW LINE.

Tecle SHIFT e EDIT simultaneamente. Estando a linha 60 no local de edição:

Tecle SHIFT e a tecla 8 com a seta apontando para a direita até o cursor postar-se após "=".

Tecle SHIFT e RUBOUT até eliminar as duas aspas e o sinal de igualdade.

Digite: STR$ C. A linha deverá ficar assim: 60 LET B$=B$+ STR$ C.

Tecle NEW LINE para a linha 60 voltar corrigida para seu lugar.

Tecle RUN e NEW LINE. Repare que agora a atribuição feita a A$ não é mais sublinhada. Em seu lugar a linha 60 coloca os números de contagem da linha 50, correspondentes sempre ao total de caracteres usados, incluindo os espaços entre as palavras, quando houver mais do que uma.

Verifique que, quanto maior for a atribuição feita a A$, mais demorada será a execução do programa. Para dar-lhe maior velocidade, acrescente-lhe os comandos FAST e SLOW.

COMANDOS FAST E SLOW

FAST: Aciona a modalidade rápida de funcionamento do microcomputador.

SLOW: Aciona a modalidade de funcionamento denominado lento do microcomputador, que é a normal e estabelecida automaticamente quando o mesmo é ligado. Quando o computador estiver em operação e for acionado o comando FAST, para fazê-lo voltar ao modo SLOW será necessário acionar esse comando.

Tecle BREAK e NEW LINE

Digite:
```
 25 FAST
105 SLOW
```

Tecle RUN e NEW LINE e dê algumas rodadas no programa, observando como mudou o modo e o tempo de execução do mesmo.

A linha 25 FAST faz com que, após a entrada do dado solicitado, a execução do programa passe para o modo rápido, executando as operações de processamento quatro - 4 - vezes mais rapidamente do que no modo SLOW. No modo FAST, todavia, o computador utiliza o vídeo apenas para exibição do resultado do programa, não aparecendo a execução sequencial ou animada do mesmo, como ocorre no modo SLOW. O modo FAST é utilizado quando o programa obriga a execuções demoradas de instruções complexas ou de cálculos.

A linha 105 SLOW faz com que o programa retome seu modo normal de execução. Poderá ser eliminada se se desejar manter o programa operando apenas no modo rápido. Experimente fazê-lo e note que o vídeo "piscará" durante a digitação de dados de entrada, em consequência da velocidade de operação.

Tecle BREAK, NEW e NEW LINE.

Digite:
```
10 FOR A=1 TO 704
20 PRINT "*";
30 NEXT A
```

Tecle RUN e NEW LINE. Observe a velocidade de execução do programa, que está sendo executado em SLOW, pois quando o computador é "zerado" por NEW e NEW LINE, ele passa a operar automaticamente nesse modo. Observe mais uma ou duas vezes. Depois:

Tecle NEW LINE.

Digite: 1 FAST

Tecle NEW LINE para introduzir a linha no programa.

Tecle RUN e NEW LINE. Compare agora a velocidade do programa, após a introdução do comando FAST. Repita a rodada.

Tecle NEW LINE.

Digite: 1

Tecle NEW LINE para eliminar a linha 1 do programa.

Tecle RUN e NEW LINE. Observe que, mesmo não estando mais no programa o comando FAST, o modo de operação permanece assim. Para voltar ao modo SLOW sem mudar o programa:

Tecle SLOW e NEW LINE. Em seguida:

Tecle RUN e NEW LINE. Após o programa parar:

Tecle FAST e NEW LINE. Em seguida:

Tecle RUN e NEW LINE.

Lembre-se de que SLOW e FAST, bem como outros comandos diretos, podem operar sem constarem de instruções do programa.

Tecle NEW e NEW LINE.

Experimente mais um pequeno programa em SLOW e FAST:

Digite:

```
10 PRINT "MOVIMENTO EM SLOW"
20 FOR C=1 TO 320
30 PRINT "0";          (0 em vídeo inverso)
40 NEXT C
50 PRINT "ATENÇÃO / AGORA SERÁ EM FAST"
60 FOR C=1 TO 100
70 NEXT C
80 FAST
90 FOR C=1 TO 320
100 PRINT "0";         (0 em vídeo inverso)
110 NEXT C
120 SLOW
130 FOR C=1 TO 100
140 NEXT C
150 CLS
160 GOTO 10
```

Tecle RUN e NEW LINE. Depois de comparar bem as velocidades:

Tecle BREAK, NEW e NEW LINE.

COMANDO VAL

VAL: Fornece o valor numérico de uma "string" ou de uma expressão alfanumérica, quando esse valor existe e é distinguível pelo computador.

Digite:

```
10 PRINT "DIGITE 1 NUMERO, TECLE NEW LINE"
20 INPUT A
30 PRINT ,,"PRIMEIRO NUMERO (A): ";A
40 PRINT ,,"OUTRO NUMERO, TECLE NEW LINE"
50 INPUT B
60 PRINT ,,"SEGUNDO NUMERO (B): ";B
70 LET A$=STR$ A
80 LET B$=STR$ B
90 PRINT ,,"SOMA DE A E B = ";A+B
100 PRINT ,,"SOMA DOS 2 EM STRINGS (A$+B$) = ";
    A$+B$
110 PRINT ,,"SOMA DE VAL A$ E VAL B$ = ";VAL A$
    +VAL B$
120 PRINT ,,"PARA CONTINUAR APERTE UMA TECLA"
130 PAUSE 4E4
140 CLS
150 GOTO 10
```

Tecle RUN e NEW LINE. Note que, a fim de dar uma idéia tão nítida quanto possível das funções VAL e STR$ (esta vista antes), o programa primeiramente introduz duas variáveis numéricas através das linhas 20 e 50. Nas linhas 70 e 80 transforma essas variáveis, por meio do comando STR$, em "strings", para que, através das operações aritméticas das linhas 90, 100 e 110, possa ser melhor entendida a função VAL, que trata o conteúdo numérico de uma "string" como valor, quando este pode ser entendido como tal pelo computador. Repare bem na diferença de resultados das linhas 100 e 110: se A$="1250" e B$="750", a linha 100 fornecerá o resultado "1250750", enquanto a linha 110 dará 2000 como resultado, em razão de a função VAL tratar como valores aritméticos os conteúdos de A$ e B$.

Aparentemente de pouco alcance ou finalidade, os comandos VAL e STR$, todavia, têm grande valia na programação geral e em especial

na matemática e financeira. Seu pleno alcance, todavia, só poderá ser devidamente avaliado à medida em que o usuário for avançando na utilização prática do computador.

Rode o programa algumas vezes, introduzindo valores bem variados, e analise detidamente os resultados para melhor compreensão dos comandos utilizados. Quando se der por satisfeito:

Tecle BREAK quando o cursor não estiver no vídeo. E após:

Tecle NEW e NEW LINE.

COMANDO DIM

DIM: Reserva espaço na memória do computador para conjuntos de variáveis numéricas e alfanuméricas, cuja utilização e processamento num programa são feitos com indexação, através de variáveis de controle ou contadoras. Um conjunto de variáveis indexadas é chamado de "ninho" ou "array".

Digite:

```
 10 DIM A$(5,16)
 20 DIM A(5)
 30 DIM B(5)
 40 FOR C=1 TO 5
 50 PRINT "DIGITE A MARCA DE UM CARRO"
 60 PRINT ,,"SERÃO 5. ESTE E O NUM. ";C
 70 INPUT A$(C)
 80 PRINT A$(C)
 90 PRINT ,,"DIGITE O ANO DE FABRICAÇÃO ";
100 INPUT A(C)
110 PRINT A(C)
120 PRINT ,,"DIGITE SEU VALOR EM MILHÕES CR$"
130 INPUT B(C)
140 PRINT B(C)
150 CLS
160 NEXT C
170 PRINT ,,,,TAB 6;"VEICULOS DISPONIVEIS"
180 PRINT ,,,,"VEICULO";TAB 18;"ANO";TAB 27;"PREÇO"
190 FOR C=1 TO 5
200 PRINT ,,A$(C);TAB 18;A(C);TAB 32-LEN STR$ B(C);B(C)
210 NEXT C
```


Tecle RUN e NEW LINE e atenda às solicitações do programa através do vídeo. Ao fim da execução do programa - notação 0/210 - uma tabela estará presente no vídeo.

Nas linhas 10, 20 e 30 o comando DIM permitiu o dimensionamento da tabela: 5 NOMES, 5 VALORES e mais 5 VALORES, através de A$(5,16) para os nomes e A(5) e B(5) para os valores ou dados numéricos. O número 16 colocado após o 5, na variável alfanumérica A$, reserva espaço na memória do computador para 5 nomes de até 16 dígitos cada um, que deverão ser introduzidos através do comando INPUT da linha 70. Os valores ou dados numéricos para as variáveis A e B entram através da execução das linhas 100 e 130. Para as variáveis numéricas não há necessidade de especificar a extensão em dígitos dos dados numéricos.

A solicitação de dados para a tabela, feita pelo computador, depende do controle da variável contadora C, através do laço ou "loop" feito pelas linhas 40 e 160.

Após a entrada de dados, as linhas 170 e 180 "rotulam" a tabela, que passa a ser impressa pela execução das linhas 190, 200 e 210, também sob controle da variável contadora C.

Note que em todas as instruções de entradas de dados e exibição dos mesmos a variável contadora C está vinculada (entre parênteses), a fim de manter os três elementos de cada ítem - nome, ano e valor - correlacionados pelo mesmo índice.

Os mais variados dimensionamentos de tabelas podem ser feitos no TK, dependendo apenas da capacidade em "bytes" de sua memória RAM. Por exemplo, uma tabela com 200 ítens, com 2 títulos e 3 valores para cada ítem, pode ser perfeitamente dimensionada num TK 85, cuja capacidade é de 16 Kbytes, como mínimo normal, para a área de programas. Bastará introduzir as instruções DIM A$(200,15), DIM B$(200,15), DIM A(200), DIM B(200), DIM C(200). Naturalmente, as letras para as variáveis não precisam ser as primeiras do alfabeto, nem estar na ordem comum. Podem ser escolhidas N, X, Q etc., sendo necessário apenas cuidar em não dimensionar duas variáveis da mesma ordem com uma mesma letra, o que fatalmente ocasiona erro e paralisação do programa. Por outro lado, nada impede que uma variável numérica e outra alfanumérica utilizem a mesma letra, já que a segunda é diferenciada pelo carácter $, que deve acompanhá-la sempre e é facilmente distinguível pelo computador.

Experimente mudar as dimensões das variáveis da tabela do último programa para 10 ou 20, por exemplo, não esquecendo de alterar também as instruções das variáveis contadoras das linhas 40 e 190. Depois de estudado o programa:

Tecle NEW e NEW LINE.

COMANDOS PEEK E POKE

PEEK: Fornece o código decimal contido num "byte" ou endereço determinado pela instrução. Em outras palavras: é o conteúdo desse "byte", na forma de um número decimal entre 0 e 255, podendo representar uma instrução ou um carácter.

POKE: Carrega o "byte" ou endereço determinado com um código decimal entre 0 e 255, correspondente a uma instrução ou carácter.

Digite: PRINT PEEK 20000

Tecle NEW LINE e anote o número que aparece no canto superior esquerdo do vídeo. Em seguida:

Digite: POKE 20000,38

Tecle NEW LINE.

Digite: PRINT PEEK 20000

Tecle NEW LINE e note que agora o número fornecido pelo computador é o que foi carregado pelo comando POKE no endereço ou "byte" de número 20000, ou seja o código 38. Prosseguindo:

Digite: POKE 20000,62 - Tecle NEW LINE em seguida.
Digite: POKE 20001,156 - Tecle NEW LINE em seguida.
Digite: POKE 20002,215 - Tecle NEW LINE em seguida.
Digite: POKE 20003,24 - Tecle NEW LINE em seguida.
Digite: POKE 20004,248 - Tecle NEW LINE em seguida.

Repare que após pressionar NEW LINE aparece no canto inferior à esquerda do vídeo a notação 0/0, indicando que a instrução foi cumprida, não obstante não haver linha numerada de programa. Verifique agora

se a digitação feita teve alguma validade:

Digite:

```
10 FOR C=20000 TO 20004
20 PRINT PEEK C;" = CODIGO DO ENDEREÇO ";C
30 NEXT C
```

Tecle RUN e NEW LINE e note que os números que aparecem no vídeo como códigos dos endereços que os seguem são aqueles introduzidos pelo comando POKE através da digitação das cinco linhas anteriores. A verificação foi feita pelo comando PEEK colocado na linha 20 do pequeno programa feito para permitir uma visualização de conjunto dos cinco registros feitos pela instrução ou comando POKE.

O comando POKE permite colocar os códigos de funcionamento e os dos caracteres do microcomputador diretamente nos seus diversos "endereços" ou "bytes", diferentemente da linguagem BASIC, que se vale do interpretador existente no computador para efetuar a mesma operação, necessária para seu desempenho. Para entender melhor o quê e quais os endereços são no microcomputador, leia o capítulo ORGANIZAÇÃO DA MEMORIA DO TK 85 E AS VARIAVEIS DO SISTEMA.

Como vimos, o comando PEEK verifica e informa, através do comando PRINT, quais são os códigos contidos nos "bytes" indicados.

Através das cinco linhas digitadas acima com o comando POKE introduzimos no computador um pequeno programa em linguagem de máquina, o qual, utilizando rotinas residentes no microcomputador - que não dependem de programação a ser feita pelo usuário - imprime no vídeo um painel que o "enche" completamente com um carácter escolhido entre os seus diversos.

Como dissemos, é um pequeno programa em linguagem de máquina e, como tal, não pode ser rodado pelo comando RUN da linguagem BASIC. Para rodá-lo, faz-se necessária a utilização do comando USR.

Outro fato a ser mencionado é que programas em linguagem de máquina não são feitos com linhas de instruções numeradas, como os vistos até agora, em BASIC. A programação em linguagem de máquina é feita colocando-se diretamente nos endereços da RAM, que será vista mais adiante, ou nos "bytes" da mesma, os códigos de funcionamento e de estruturação de programas desse tipo.

COMANDO USR

USR: Permite acesso direto a qualquer "endereço" ou "byte" das memórias ROM e RAM do computador, para execução de subrotinas ou programas em linguagem de máquina.

Rode agora o pequeno programa feito em linguagem de máquina:

Digite: RAND USR 20000

Tecle NEW LINE e admire o painel no vídeo. Em seguida:

Digite: POKE 20001, 134 - Tecle NEW LINE em seguida.

Digite: RAND USR 20000 - Tecle NEW LINE em seguida.

Experimente depois "POKEAR" o mesmo endereço 20001 com códigos como 5, 23, 61, 219 e outros. Não mexa nos endereços 20000, 20002, 20003 e 20004, sob pena de o computador entrar em confusão, "saindo do ar", e ter que ser desligado e religado, com a perda do programa.

Programas em linguagem de máquina são sempre críticos, por menores que sejam, e qualquer pequeno erro pode pôr o programa a perder facilmente, quase sempre fazendo o computador "sair do ar".

Tecle NEW e NEW LINE.

O programa apresentado a seguir, utilizando também os comandos POKE, PEEK, RAND e USR, ilustra outra maneira de introduzir instruções em linguagem de máquina, através de códigos hexadecimais, quase sempre utilizados para essa finalidade. O programa introduz também outros recursos, cuja análise será de valia para utilizações futuras:

DIGITE:

```
10 LET A=20000
20 LET B=1
30 LET C=2
40 LET A$="3E9CD718F8ST"
50 PRINT ,,,,"DIGITE O CODIGO HEXADECIMAL: ";
   A$(B);A$(C)
60 PRINT ,,"(A SEGUIR TECLE NEW LINE)"
70 INPUT A$
```



```
80 IF A$="ST" THEN GOTO 190
90 POKE A,CODE A$*16+CODE A$(2)-476
100 LET D=PEEK A
110 PRINT ,,CHR$ (INT (D/16)+28);CHR$ (D-INT (D/16)*
    16+28);" = HEXADEC. EQUIV. AO DEC. ";D
120 FOR I=1 TO 120
130 NEXT I
140 LET A=A+1
150 LET B=B+2
160 LET C=C+2
170 CLS
180 GOTO 40
190 CLS
200 PRINT "ST = SUSTA ENTRADA DE CODIGOS"
210 PRINT "VOCE INTRODUZIU UM PROGRAMA EM"
220 PRINT "LINGUAGEM DE MAQUINA A PARTIR DO"
230 PRINT "ENDEREÇO 20000. PARA RODA-LO TE-"
240 PRINT "CLE RAND USR 20000 E NEW LINE."
250 PRINT ,,"DEPOIS VERIFIQUE SE COMPREENDEU."
```

Tecle RUN e NEW LINE. A medida que o programa vai sendo executado, proceda de acordo com as instruções que o mesmo coloca no vídeo. Depois rode-o mais algumas vezes com o fito de estudá-lo.

Observe bem a ação da linha 90, cujo efeito é demonstrado através da função da linha 110.

Tente compreender como a "string" da variável A$ é "fatiada" pela função da linha 50, com auxílio das variáveis B e C, cuja incrementação é feita nas linhas 150 e 160 para efeito de prosseguir com o fatiamento, que é um recurso muito interessante do computador para manipular "strings".

Tecle NEW e NEW LINE.

Digite

```
1 REM .........................
  ...................          (45 pontos)
10 LET E=16514
20 INPUT E$
30 IF E$="S" THEN GOTO 100
40 POKE E, CODE E$*16+CODE E$(2)-476
```

```
 50 LET D=PEEK E
 60 SCROLL
 70 PRINT CHR$ (INT (D/16)+28);CHR$ (D-INT (D/16)*
    16+28);" ";E
 80 LET E=E+1
 90 GOTO 20
100 CLS
110 PRINT ,,"DIGITE RAND USR 16546 E NEW LINE"
```

Tecle RUN e NEW LINE. O CURSOR L no canto inferior esquerdo do vídeo estará indicando que é aguardada entrada de dados.

Digite os códigos hexadecimais que se seguem, um de cada vez, teclando NEW LINE após cada par digitado. Espere sempre o CURSOR L estar no seu lugar para digitar um código. Lembre-se sempre de que os códigos hexadecimais são sempre constituídos por dois dígitos.

Códigos a introduzir: 39 30 00 24 21 00 16 00 29 34 32 2E 33 2E 34 00 37 26 35 2E 29 34 00 16 00 33 28 26 38 26 37 2E 11 82 40 01 20 00 CD 6B 0B C9 S.

Se tudo foi feito certinho, deverá surgir no vídeo a frase da linha 110. Execute-a:

Digite RAND USR 16546 e tecle NEW LINE.

Não se admire de ter surgido no vídeo apenas o título deste livro e o nome de seu autor.

Tecle NEW LINE e observe como ficou a linha 1 REM que foi digitada com 45 pontos. Lá estão todos os símbolos dos códigos hexadecimais digitados através do programa em BASIC. A linha 1 é agora um programa em linguagem de máquina, completo, embora pequeno.

Delete a seguir o programa em BASIC, deixando apenas a linha 1. A deleção ou eliminação é feita, como já vimos antes, digitando-se os números das linhas do programa, um de cada vez, e teclando NEW LINE em seguida. Depois de feita a deleção:

Digite: RAND USR 16546

Tecle NEW LINE e constate como o programa continua completo e independente.

Esse pequeno programa em linguagem de máquina - 42 códigos hexadecimais (3 sobraram de reserva) não conseguem fazer um grande programa, realmente - foi elaborado com a finalidade de demonstrar sucintamente a estrutura básica desse tipo de programação, a fim de suscitar algumas considerações:

Se for consultada a tabela de caracteres do TK, será visto que a letra T tem o código hexadecimal 39, a letra K o código 30, o espaço o código 00, o hífen o 16 etc. Assim, para escrever em linguagem de máquina a frase toda, contando também os espaços entre as palavras, código 00, foi necessária a utilização de 32 códigos. Os restantes, 10, constituem as instruções para produzir no vídeo a frase. E note que todos os códigos tiveram que ser colocados em "endereços" consecutivos a partir de um predeterminado, condição imprescindível para um programa em linguagem de máquina.

O programa em BASIC das linhas 1 a 110 serviu apenas para introduzir os códigos hexadecimais através do comando POKE da linha 40, que transforma os mesmos na codificação inteligível para o computador. Poderia o mesmo programa ter sido feito para introduzir códigos decimais equivalentes, com alguma simplificação. Ocorre, todavia, que os comandos, instruções e funções em linguagem de máquina são geralmente programados em códigos hexadecimais, tirados da lista de códigos estabelecidos pelo fabricante do microprocessador, que assim fornece a equivalência com o sistema binário, que efetivamente comanda o funcionamento cérebro-eletrônico do computador, mas cuja manipulação e inteligibilidade seriam muito pouco acessíveis à grande maioria dos usuários.

De forma geral, a única vantagem dos programas elaborados em linguagem de máquina é a velocidade de processamento, principalmente para jogos e alguns tipos de programas complexos, mas não há dúvida de que sua utilização apresenta dificuldades consideráveis, sendo praticamente inacessível se não for exaustivamente estudada e decorada, ao contrário da linguagem BASIC, chamada de alto nível, e realmente inteligente e extremamente inteligível.

A frase utilizada no pequeno programa em linguagem de máquina e a título de exemplo poderia ser produzida no vídeo, em BASIC, com um

simples e rápido comando: PRINT "TK 85 - DOMINIO RAPIDO - NELSON CASARI" e NEW LINE.

Deve ser dito, entretanto, que tanto quanto for melhorada a interação usuário-computador, a linguagem de máquina deve ser aprendida e aplicada, principalmente visando obter os melhores rendimentos do fabuloso e genial aparelho.

COMANDOS COPY, LLIST E LPRINT

São utilizados somente quando o computador estiver acoplado a uma impressora. Se acionados sem a mesma, nada fazem.

COPY: Envia para a impressora o que estiver na área de memória de vídeo do computador, ou, em outras palavras, o que for exibido pelo computador na tela de vídeo, em consequência da execução de um programa.

LLIST: Funciona como LIST, enviando a listagem do programa para a impressora.

LPRINT: Funciona como PRINT, usando a impressora em vez da tela de TV.

COMANDOS SAVE E LOAD

SAVE: Faz com que um programa existente na memória do computador seja gravado numa fita K-7, através de um gravador K-7 comum.

LOAD: Faz com que um programa previamente gravado em fita padrão K-7 seja carregado na memória do computador através de um gravador K-7 comum.

Deixamos propositadamente para o fim deste capítulo a introdução do uso destes comandos, em razão de sua estreita ligação com um complemento dos mais importantes na utilização do microcomputador: o gravador K-7 e, consequentemente, o armazenamento de programas e dados em fita K-7, assuntos que serão tratados nos capítulos seguintes a este e cujo conhecimento e completo domínio devem constituir-se num dos principais objetivos do usuário.

Vamos, pois, abordar tais questões com a plenitude possível, para que sejam evitados certos problemas que costumam ocorrer.

DESCERRANDO SAVE E LOAD

Vejamos primeiramente SAVE que, detalhando, é o comando através do qual os dados de um programa existente no computador são transformados em sinais acústicos passíveis de serem gravados em fita K-7 comum por intermédio de um gravador K-7 também comum, estando ambos - microcomputador e gravador - conectados através das respectivas tomadas MIC existentes em ambos, e com auxílio de um cabo coaxial padrão.

Procedimento: Havendo um programa na memória do computador e estando o gravador preparado para gravar, isto é, conectado ao computador conforme descrito acima, com fita K-7 de boa qualidade, de preferência virgem, as teclas REC e PLAY do mesmo devem ser acionadas, devendo ser logo em seguida acionado o comando SAVE e pressionada a tecla NEW LINE do computador. Atenção! O comando SAVE deverá ser digitado com qualquer identificação, isto é, deverá ser seguido de um nome, entre aspas, de preferência o do próprio programa, sem o que não funcionará. Exemplo: SAVE "PING PONG". Essa identificação poderá ser utilizada depois, quando o programa tiver que ser carregado da fita para o computador, para que este procure e selecione o programa na fita, caso a mesma tenha diversos programas gravados e não tenha sido anotada sua localização exata.

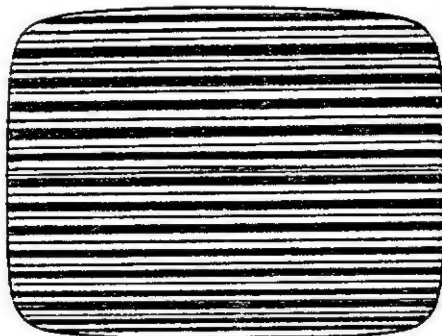
Os gravadores atuais normalmente funcionam com nível automático de gravação, dispensando qualquer ajuste. Por precaução, apenas o controle de tom deverá ser firmado no ponto máximo de reprodução aguda.

Iniciado o processo, a tela de TV ficará acinzentada por aproximadamente cinco segundos e em seguida aparecerão na mesma listras características do processo e que indicam que o programa está sendo mandado para o gravador.

As listras se movimentam sucessiva e rapidamente, sendo acompanhadas de som característico reproduzido pelo alto-falante da TV, som que logo se tornará familiar e facilmente reconhecível.

Quando a transmissão de todo o programa terminar, as listras e

som característicos cessarão e aparecerá no vídeo, no canto inferior esquerdo, a notação 0/0, indicando o término do processo. Apenas deverá ser desligado o gravador. O programa estará gravado na fita K-7.

A ilustração apresentada a seguir dá uma idéia de como a tela de TV se apresenta durante o processo de gravação de um programa:

SAVE - IMAGEM DE VIDEO

O programa transmitido para o gravador ainda permanecerá no computador. Poderá ser rodado normalmente, usado para outra gravação ou apagado pelo processo usual. Se for apenas pressionada a tecla NEW LINE ele surgirá novamente listado no vídeo.

Esse é o caso de um programa normal e comum, no qual não exista uma linha de instrução SAVE.

No caso de programas em que conste uma linha dessa ordem, o comando para SAVE deverá ser feito através de GOTO. Exemplificando: se o programa tiver uma linha:

2500 SAVE "PING PONG"

o comando para gravar deverá ser: GOTO 2500 e NEW LINE, naturalmente após o gravador ter sido acionado em REC e PLAY.

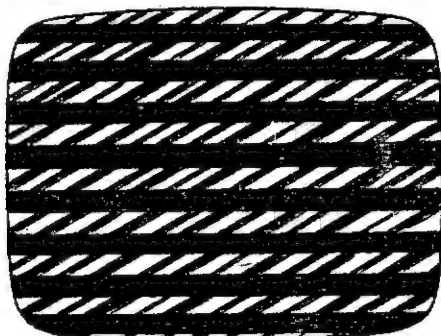
Quando existe o comando SAVE numa linha de instrução do programa é quase absolutamente certo que a mesma seja seguida de uma linha de instrução RUN, o que fará com que o programa, após ser gravado, seja automaticamente posto a rodar pelo computador. Em tal caso, para gravá-lo outras vezes, deverá ser acionado um dos comandos BREAK ou STOP - o que melhor se aplicar - acionando novamente a seguir o comando GOTO mais número de linha e NEW LINE, não esquecendo que o gravador deverá estar funcionando na modalidade de gravação.

Como sabemos, LOAD é o processo que permite carregar, ou melhor, transferir para o microcomputador um programa previamente gravado em fita K-7, por intermédio de um gravador K-7 comum, estando ambos - microcomputador e gravador - conectados através das respectivas tomadas EAR próprias a ambos e com auxílio de um cabo coaxial comum.

Procedimento: A fita K-7 com o programa a ser transferido para o microcomputador deverá ser posicionada no gravador com algum intervalo antes do ponto inicial da gravação. O controle de volume do gravador deverá ser ajustado entre metade e três quartos do máximo e o controle de tom colocado na posição de reprodução máxima de sons agudos. O comando LOAD deverá ser digitado seguido do nome do programa colocado entre aspas, caso a fita não possa ser colocada de maneira a que o ponto de início do programa esteja próximo e antes do cabeçote de leitura do gravador e tiver que ser localizado na fita pelo computador. Ou digitado apenas seguido de aspas duplas quando o programa a ser carregado é o primeiro após o ponto em que se encontra a fita. Exemplificando, para melhor compreensão: LOAD "PING PONG" ou simplesmente LOAD"" (aspas da tecla P). Em seguida, deverá ser pressionada a tecla NEW LINE, deixando o gravador parado, isto é, sem acionar de imediato a tecla PLAY.

Iniciado no microcomputador o processo, surgirão no vídeo riscas ou listras características, diferentes do processo SAVE, que se movimentarão rapidamente no sentido diagonal à direita, como se percorressem a tela de alto a baixo.

Deverá então ser acionada a tecla PLAY do gravador e, tão logo seja atingido o ponto de início de gravação, será reproduzido no alto-falante da TV um som característico do processo de carregamento, diferente do processo SAVE, e as listras do vídeo assumirão aspecto diverso do inicial. Estas também aparentarão movimento rápido e sucessivo, como se tremessem, e deverão permanecer assim enquanto não se completar o processo. Ao término do mesmo, quando cessarem, deverá aparecer no canto inferior esquerdo do vídeo a notação 0/0, indicando que o programa "entrou" na memória do computador. O gravador deverá ser desligado e o programa poderá ser manipulado normalmente no computador.

A ilustração apresentada a seguir dá uma idéia de como a tela de TV se apresenta durante o processo de carregamento de um programa:

LOAD - IMAGEM DE VIDEO

Se o programa carregado tiver uma linha de instrução SAVE seguida de outra com o comando RUN, entrará "rodando" no computador.

Se, durante o processo de LOAD, cessarem as listras no vídeo sem aparecer em seguida a notação 0/0, não tendo também entrado o programa rodando, o processo não se completou com sucesso, devendo ser reiniciado, o que poderá ser feito se aparecer o CURSOR K no canto inferior esquerdo do vídeo. Caso não apareça e o vídeo permanecer acinzentado, o microcomputador deverá ser desligado e religado antes de o processo de carregamento ser reiniciado, sendo aconselhável, neste caso, proceder a novo ajuste do controle de volume do gravador, posicionando-o pouco acima da posição anterior. Se o volume do gravador estiver abaixo do mínimo necessário ou muito acima do mesmo, geralmente ocorre insucesso no carregamento.

GRAVADOR K-7, A OUTRA METADE DO MICROCOMPUTADOR

Embora utilizada aqui como força de expressão, a afirmação encerrada pelo título acima corresponde quase que inteiramente com a realidade. Apenas a proporção deverá ser retificada, talvez para 1/3, pouco menos, já que um microcomputador se torna praticamente imprestável sem um aparelho de TV como monitor, pouco praticável sem um gravador, periféricos implementares indispensáveis.

Os microcomputadores geralmente têm memórias limitadas a níveis práticos e econômicos, fato muito mais acentuado com os chamados de uso pessoal. Não seria bom senso fabricá-los com memórias agigantadas e complexas, uma vez que os programas normalmente destinados ou exigidos pela maioria dos usuários se caracterizam muito mais por diversificação do que por extensão.

Sua estruturação não permite a memorização e manipulação de múltiplos programas extensos a um só tempo, o que não invalida a necessidade de tê-los e processá-los separadamente e que patenteia a necessidade do gravador K-7.

Certos tipos de microcomputadores mais sofisticados, e mesmo alguns considerados impropriamente de "uso pessoal", são ou podem ser implementados por DISC-DRIVERS em lugar dos gravadores K-7 comuns, com a vantagem em velocidade de processamento e localização mais rápida dos programas contidos num disco, mas a um custo muito mais elevado, consideravelmente mais elevado, que só seria justificado por uma utilização bastante ampla e eventualmente profissional.

Os microcomputadores atuais produzidos no Brasil, exemplos exponenciais dos quais o TK 85 e o CP 200, incorporam duas velocidades de processamento de fitas K-7: a normal, relativamente lenta, 300 "bauds" por segundo, e a ultra-rápida, denominada HIGH SPEED, de 4200 unidades equivalentes por segundo. Nesta modalidade, um programa de 16 K bytes pode ser gravado ou carregado em 30 segundos, o que é realmente muito satisfatório.

Praticamente qualquer gravador K-7 pode ser adotado como complemento periférico de um microcomputador, para utilização na modalidade de velocidade normal, não se recomendando, todavia, aqueles que necessitam de conversor externo, para substituir as pilhas, em razão de que a maioria dos conversores gera ruídos que são captados pelo gravador e põe a perder a gravação ou o carregamento. Seu uso é simples e prático e exige apenas alguns cuidados especiais:

1º - Ajuste adequado dos cabeçotes, especialmente do de leitura, que efetua a gravação e a reprodução, com limpeza e desmagnetização periódicas. Em certas circunstâncias o ajuste desse cabeçote deve ser alterado para que possa ser efetuada com sucesso a leitura de uma fita gravada em aparelho com regulagem de cabeçote desigual.

2º - Uso e manutenção apropriados, para que as folgas do mecanismo não excedam as tolerâncias de forma a comprometer a gravação e reprodução de um programa.

3º - Não alimentar o gravador com pilhas quando em uso com o computador, em razão de que o desgaste relativamente rápido das pilhas provoca alterações nas rotações do motor do gravador, pondo a perder a gravação ou o carregamento.

Já a utilização na modalidade de velocidade ultra-rápida ou HIGH SPEED exige que o gravador seja dotado de cabeçote de gravação e reprodução com alta relação sinal-ruído e circuito eletrônico com faixa ampla de passagem de frequências altas - responsáveis pelos sons agudos - características aliás consideradas apenas normais nos aparelhos considerados de primeira qualidade, conquanto comuns, o que, todavia, não anula a probabilidade de alguns tipos ou marcas não se prestarem para uso nesta modalidade.

De qualquer forma, se um gravador K-7 comum de boa qualidade não se prestar para utilização na modalidade HIGH-SPEED, é praticamente garantido que qualquer rádio-gravador, não estéreo, o faça, naturalmente usado apenas como gravador-reprodutor K-7.

COMO AJUSTAR O CABEÇOTE DO GRAVADOR

A ilustração mostrada a seguir apresenta a disposição de fixação dos cabeçotes de um gravador:

No gabinete do aparelho e próximo da tampa do compartimento alojador da fita, exatamente acima do parafuso de regulagem do cabeçote - quando este se encontra deslocado para a frente pelo acionamento da tecla PLAY - costuma haver um pequeno orifício que permite o acesso de uma pequena chave de fenda à cabeça do citado parafuso.

Quando não existir esse orifício, deverá ser feito com o auxílio de uma pequena broca e com cuidados apropriados.

Procedimento para ajuste:

1º - O gravador deverá estar desconectado do microcomputador.

2º - Colocar no gravador uma fita K-7 com programa gravado para computadores, de boa qualidade, de preferência produzida por firma renomada em SOFTWARE.

3º - Ajustar o controle de volume do gravador em aproximadamente 1/3 de sua capacidade e o controle de tonalidade no ponto máximo para reprodução de sons agudos.

4º - Acionar a tecla PLAY do gravador.

5º - Introduzir uma chave de fenda pequena no orifício referido e posicionar a mesma adequadamente na fenda do parafuso de regulagem do cabeçote.

6º - Com ouvido atento ao som reproduzido pelo gravador, girar a chave de fenda suavemente, muito pouco a cada vez, para a esquerda, depois para a direita, a fim de identificar em que posição se torna mais aguda e melhor a reprodução da gravação da fita.

7º - Adotar essa posição como padrão para gravações próprias, se possível marcando ou memorizando a exata direção da fenda do parafuso para futuras referências e retorno no caso de necessidade de eventuais alterações.

E sempre oportuno lembrar que, após 10-12 horas de trabalho efetivo, os cabeçotes do gravador devem passar por cuidadosa limpeza e, se possível, desmagnetização.

A limpeza dos mesmos deve ser feita com produtos apropriados, facilmente encontrados no comércio especializado, ou com álcool isopropílico, obtido em drogarias, com ajuda de um cotonete. Jamais devem ser usados produtos abrasivos.

A desmagnetização deverá ser feita com desmagnetizador ou fita K-7 produzida para esse fim, também facilmente encontrados no comércio.

OS CONTROLES DE VOLUME E TOM DO GRAVADOR

Outro ponto que deve merecer atenção a fim de ser obtido rendimento otimizado do gravador, principalmente para fins de carregamento de programas da fita K-7 para o microcomputador, é o que diz respeito aos controles de volume e de tom do gravador e suas posições.

Como já foi dito, o controle de tom deverá estar sempre no ponto de limite máximo para reprodução de sons agudos.

O controle de volume deverá ir sendo experimentado a partir dum ponto pouco acima do médio de sua capacidade, até o ponto a partir do qual os programas sejam carregados no microcomputador sem obstáculos.

Se o botão de controle de volume tiver alguma marca que permita uma referência em relação à janela em que gira, deve ser memorizada a sua exata posição no ponto acertado, para ser posicionado ali sempre antes do início do processo de carregamento. Se não tiver nenhuma marca, esta poderá ser feita com esmalte de unhas de cor clara, com auxílio de um palito de dentes, numa das ranhuras do botão, a mais próxima possível do ponto de melhor visualização para o usuário, a fim de facilitar o ajuste certo sempre que necessário.

Deve ser lembrado que, durante o processo de carregamento de um programa, se a imagem no vídeo e o som característicos indicarem normalidade, as posições dos controles do gravador não devem ser alteradas, sob pena de o processo ser interrompido.

Finalmente, quando possível, ou pelo menos uma vez ao ano, os resistores internos ou grafites desses controles, sobre os quais desliza o cursor divisor de tensão acionado pelo botão, devem ser lubrificados com produto apropriado, geralmente vendido sob a forma de "spray". Esse serviço poderá ser feito em alguns minutos por um técnico especializado, ou mesmo pelo usuário que saiba como fazê-lo, e evitará problemas eventuais de contacto que poderão prejudicar ambos os processos de gravação ou carregamento da fita K-7.

UM PROGRAMA PARA SAVE E LOAD

Apresentamos a seguir um pequeno programa que, além de servir inicialmente para fins de experiência e aquisição de prática no domínio das funções ligadas ao uso conjunto do microcomputador e gravador, poderá depois ser usado como padrão para verificação das condições de entrada ou carregamento de programas outros e ajustes do cabeçote gravador-reprodutor sempre que necessário. Bastará ter sempre disponível uma fita com algumas gravações do pequeno programa, que deverão ser feitas como indicamos a seguir.

Digite:

```
10 SCROLL
20 PRINT "CARREGAMENTO OK"
30 SCROLL
40 PRINT TAB 16;"PRESSIONE BREAK"
50 GOTO 10
60 SAVE "TESTE"
70 GOTO 10
```

Não obstante já termos detalhado os procedimentos de SAVE e LOAD em capítulo anterior, vamos repassá-los, agora na prática, visando dirimir dúvidas ou dificuldades, já que esta parte da microcomputação é geralmente causa de problemas desgastantes para o usuário de menor experiência.

Procedimento:

1º - Conecte o cabo coaxial apropriado nas tomadas MIC do microcomputador e do gravador.

2º - Coloque no gravador uma fita K-7 de boa qualidade, de preferência virgem, cuidando para que a mesma esteja no ponto inicial.

3º - Apenas por precaução, ajuste o controle de tonalidade para reprodução máxima de sons agudos.

4º - Acione as teclas REC e PLAY do gravador.

5º - Digite no micro: GOTO 60 e tecle NEW LINE.

6º - O vídeo se tornará acinzentado e dentro de poucos segundos surgirão no mesmo as listras características do processo SAVE, conforme ilustração anterior. Logo as mesmas cessarão e o pequeno programa teste estará "rodando" no vídeo.

7º - Acione a tecla STOP do gravador e em seguida pressione a tecla BREAK do microcomputador.

O programa foi gravado uma vez. Faça agora a segunda gravação a partir do ponto em que parou a fita:

1º - Acione REC e PLAY no gravador.

2º - Digite no micro: GOTO 60 e tecle NEW LINE.

3º - Observe o vídeo e atente para o som característico.

4º - Quando o programa aparecer "rodando" no vídeo, acione a tecla STOP no gravador e BREAK no microcomputador.

O programa foi gravado pela segunda vez. Faça ainda mais duas ou três gravações, repetindo os últimos quatro ítens do procedimento.

Em seguida, passe o cabo coaxial para as tomadas EAR do microcomputador e do gravador e proceda como se segue:

a) - Rebobine a fita K-7 no gravador, colocando-a no ponto inicial, que é a parte colorida ou transparente da fita.

b) - Tecle NEW e NEW LINE no microcomputador.

c) - Ajuste o controle de volume do gravador no ponto entre 2/3 e 3/4 do máximo e o controle de tom no ponto máximo.

d) - Digite no micro: LOAD "TESTE" ou simplesmente LOAD "" e tecle NEW LINE.

e) - No vídeo deverão surgir listras características do processo LOAD, movimentando-se no sentido diagonal à direita.

f) - Acione a tecla PLAY do gravador.

g) - Quando for atingido o ponto de início da gravação na fita as listras assumirão o aspecto característico do processo de carregamento, conforme ilustração já vista, e o som também característico será reproduzido pelo alto-falante da TV.

h) - Logo as listras cessarão e surgirá o programa teste "rodando" no vídeo.

i) - Acione a tecla STOP do gravador.

j) - Proceda normalmente com o programa no microcomputador.

Caso o carregamento não se realize, isto é, não sejam cumpridos os passos até o final com sucesso, repita o processo, a partir do ponto em que parou a fita - portanto antes da segunda gravação - aumentando um pouco mais o volume de reprodução do gravador. Caso este já esteja muito próximo do ponto máximo pode ter sido essa a causa da falha. Diminua o volume, então, colocando o controle pouco acima do ponto correspondente a 2/3 de seu curso.

Mesmo depois de efetuado o carregamento com sucesso, repita ainda várias vezes o processo todo de LOAD, atento à imagem de vídeo e ao som, para familiarização mais rápida com o mesmo.

Uma observação ainda a ser feita: lembrar que todo programa que tenha o comando SAVE como linha de instrução, a exemplo do programa teste acima, entra em execução automaticamente e imediatamente após completar-se o processo de carregamento no microcomputador.

GRAVAÇÕES E CARREGAMENTOS EM HIGH-SPEED

Os procedimentos, de forma geral, são os mesmos de SAVE e LOAD, diferindo apenas quanto aos comandos:

Em vez de SAVE "NOME"

Digite: RAND USR 8405 e tecle NEW LINE
(Corresponde a HI-SAVE)

Em vez de LOAD "NOME" ou LOAD ""

Digite: RAND USR 8630 e tecle NEW LINE
(Corresponde a HI-LOAD)

A modalidade HIGH-SPEED permite também conferir a gravação feita de um programa existente na memória do computador. Seu comando é:

RAND USR 8539 e tecle NEW LINE
(Corresponde a HI-VERIFY)

O uso da função HI-VERIFY deve ser feito logo após ter sido feita uma gravação de um programa existente na memória do computador.

O programa do qual foi feita a gravação deverá ser mantido normalmente no computador e deverá ser adotado o procedimento enumerado a seguir:

- a fita em que foi feita a gravação deverá ser rebobinada até o ponto um pouco anterior ao inicial da gravação feita;

- devem ser conectados pelo cabo coaxial as tomadas EAR do computador e do gravador;

- o volume do gravador deve ser ajustado pouco abaixo da metade de seu curso. (Na modalidade HIGH-SPEED o volume para reprodução da fi

ta deve ser menor do que na modalidade normal.)

- deve ser acionado o comando RAND USR 8539 e NEW LINE no computador, sendo em seguida posto o gravador a funcionar em PLAY.

Quando cessarem na TV os sinais característicos do processo de carregamento, surgirá no vídeo a notação "OK" se a gravação estiver perfeita, ou "ERRO" se estiver imperfeita.

No caso de surgir a notação "ERRO", repita o processo, alterando ligeiramente o nível de volume do gravador, se não estiver ainda familiarizado com o processo e não tiver ainda determinado o nível ideal de volume para esta modalidade.

Se persistir a notação de "ERRO", faça nova gravação e repita todo o processo, sempre utilizando o programa existente na memória do microcomputador.

O nível de volume para carregamentos ou verificações em HIGH-SPEED, respectivamente HI-LOAD e HI-VERIFY, difere do de carregamento em modalidade normal, LOAD, bem como diferem as imagens e sons gerados por todo o processo, incluindo HI-SAVE. Mas não há dúvidas de que algumas poucas observações logo tornarão o usuário familiarizado com os referidos, especialmente se já estiver habituado com os processos normais de gravação (SAVE) e carregamento (LOAD).

De qualquer maneira, será sempre aconselhável fazer várias experiências no início, a fim de estabelecer o nível ideal de volume para o aparelho que será usado normalmente, já que não é incomum ocorrerem diferenças até em aparelhos idênticos ou de uma mesma marca.

UM IMPLEMENTO ÚTIL PARA GRAVAÇÕES E CARREGAMENTOS DE FITAS K-7

Vimos antes que, para transmitir um programa do microcomputador para uma fita K-7, isto é, gravar em fita um programa que esteja na memória do computador - através de SAVE - é preciso conectar as respectivas tomadas MIC de ambos com cabo coaxial provido de plugues nas extremidades, o mesmo acontecendo quando se quer carregar no microcomputador um programa previamente gravado em fita K-7 - através de LOAD - sendo a conexão feita, neste caso, pelas respectivas tomadas EAR de cada aparelho, com auxílio do mesmo cabo coaxial.

O uso permanente de dois cabos ligando as quatro tomadas não se recomenda em vista da grande probabilidade de ocorrência de realimentação dos sinais acústicos, com consequente prejuízo no processamento de gravação ou carregamento.

Desconectar e conectar alternativamente os cabos, ou trabalhar com um único cabo em situação idêntica, também não constitui procedimento cômodo, além de aumentar as probabilidades de engano nas conexões, principalmente em momentos em que toda a atenção é dispensada a aspectos intrínsecos e mais importantes do trabalho com o computador, devendo ser considerada, ainda, a possibilidade de a repetição constante desse procedimento causar danos às tomadas dos aparelhos, quando não curtos-circuitos - apenas danosos, não perigosos no caso - em consequência do estrago eventual do cabo conector.

Uma solução que parece impor-se pela simplicidade, baixo custo e praticidade, é a adoção do implemento descrito a seguir, cuja realização é extremamente exequível, passando a ser desempenhada pelo mesmo, através do simples acionamento de um ou outro botão-tecla que integram o conjunto, a tarefa de comutação das conexões necessárias para LOAD e SAVE.

Basicamente, o pequeno implemento é constituído por duas chaves reversoras funcionando como interruptoras de dois cabos coaxiais providos de plugues nas extremidades e que fazem as conexões EAR e MIC

do microcomputador e do gravador.

O mesmo é munido também de um pequeno alto-falante para monitoração da reprodução de fitas, o que elimina a necessidade de desplugar-se a tomada EAR do gravador quando se quer verificar uma gravação ou localizar o seu início ou o seu fim numa fita.

Um potenciômetro incorporado ao conjunto permite controlar o volume de som de monitoração durante a reprodução da fita, sem que haja necessidade de alterar-se o controle de volume do gravador, que poderá ser mantido na posição ideal de funcionamento para a finalidade de carregamento, posição essa determinada pela experiência ou com auxílio do monitor visual constituído pelos dois pequenos LEDS - díodos luminescentes - integrantes do conjunto.

Ao receber tensão acima de 0,2 volts um LED passa a emitir luz - vermelha, verde ou amarela, dependendo do tipo adotado - atingindo seu ponto máximo de luminiscência quando a tensão atingir a 1,5 volts.

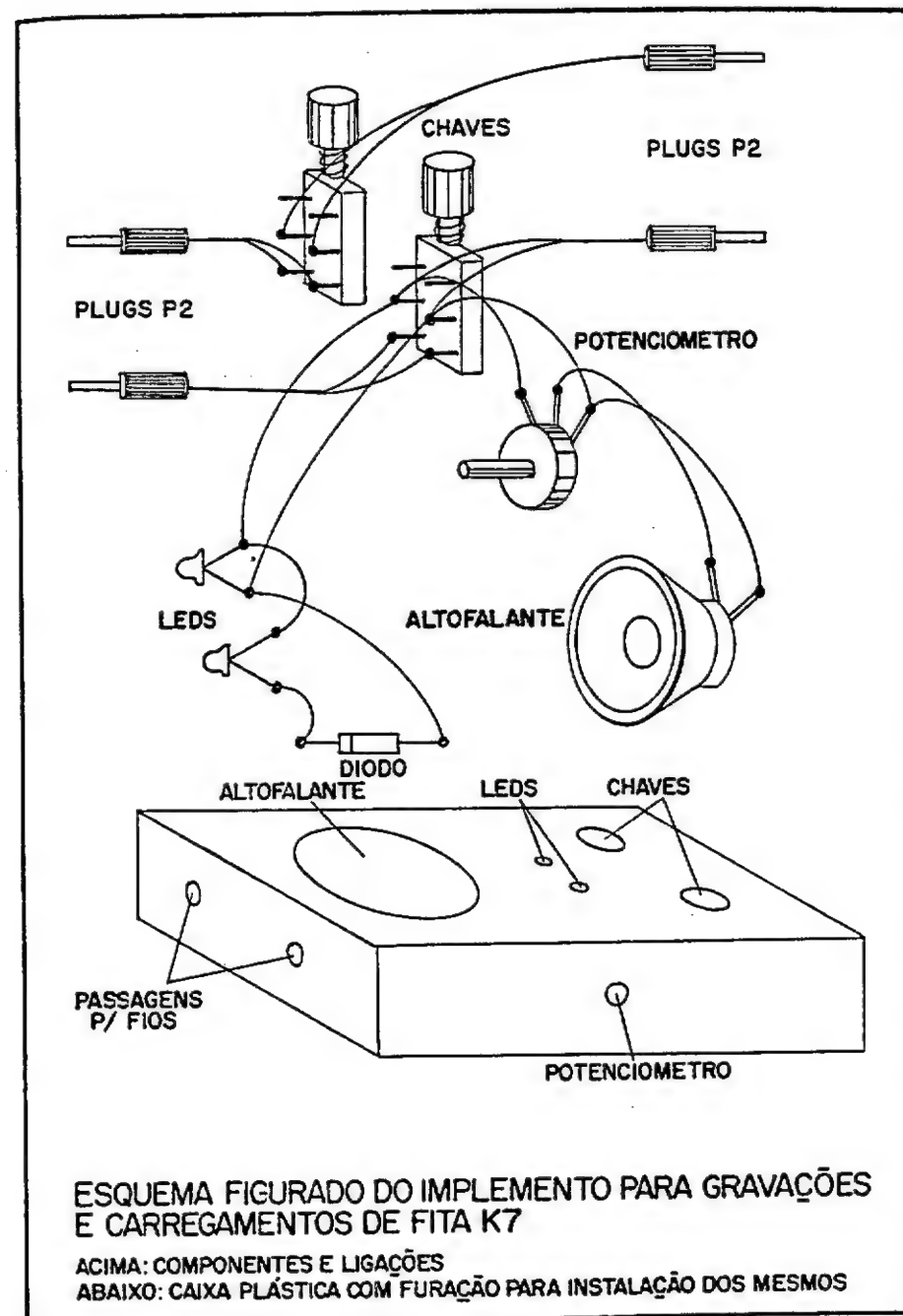
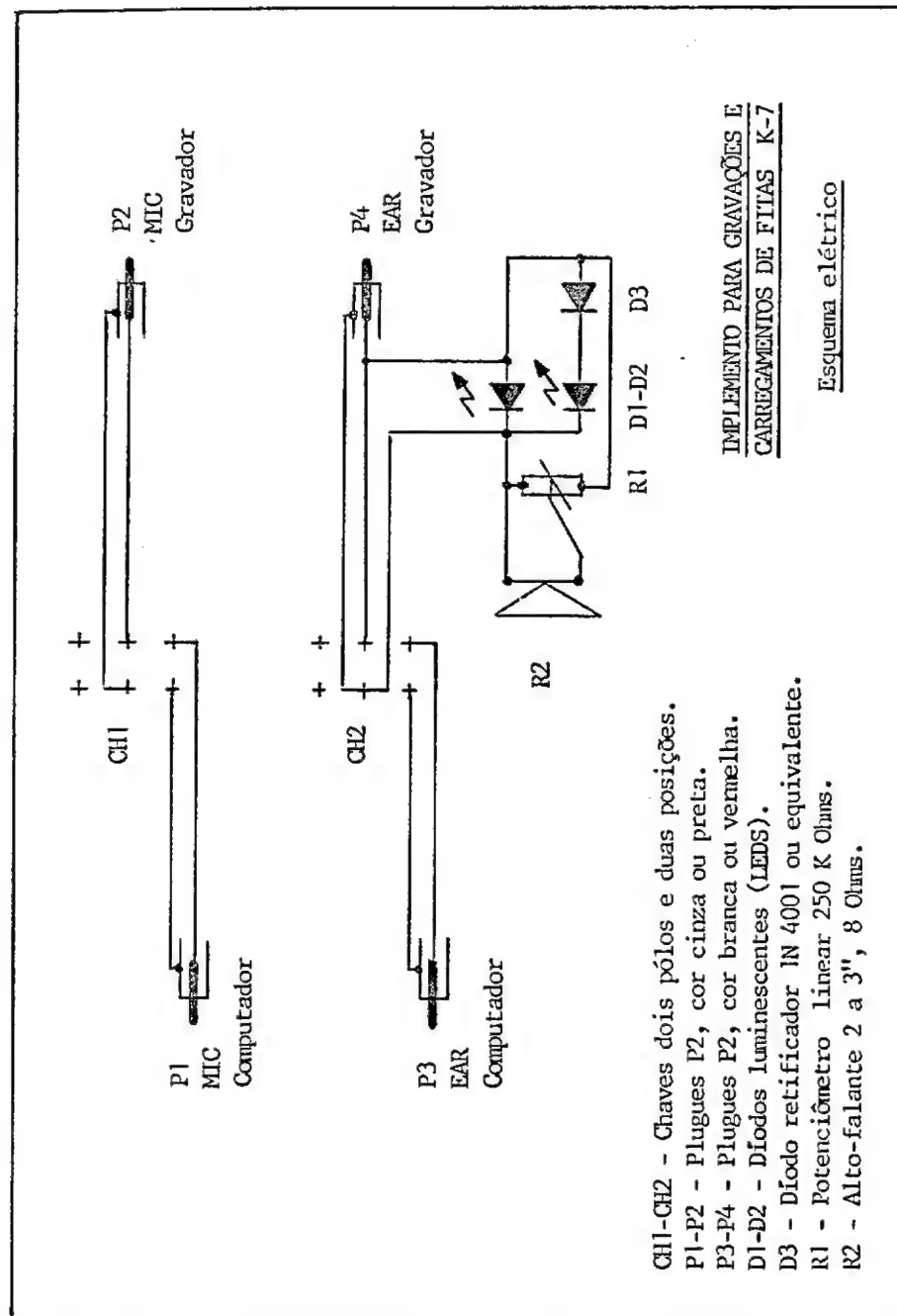
Da maneira como são ligados os dois leds do implemento, quando a potência de som produzido pelo gravador atinge a marca aproximada de 1,6 volts, um dos leds produz sua luminosidade plena enquanto o outro começa apenas a bruxulear. Um ligeiro aumento de volume do gravador em relação a esse ponto, quando então o segundo led passa a produzir aproximadamente 20% de sua luminosidade, deve ser o ponto ideal para se efetuar o carregamento de um programa de uma fita K-7 para o computador, já que a tensão requerida pelo computador para essa finalidade gira em torno desse valor.

Assim, os leds funcionam como indicadores visuais durante o processo de carregamento ou LOAD.

A montagem do implemento é bastante simples, podendo ser feita em qualquer caixa plástica de tamanho adequado, cuidando apenas de escolher bem os locais para alojamento dos componentes para, em seguida, executar os furos e cortes necessários.

São apresentados nas páginas seguintes dois esquemas do implemento, um elétrico e outro figurado, com indicações dos componentes utilizados. Acreditamos ser realmente fácil, com base nos mesmos, montar este utilíssimo implemento, que não se encontra à venda no mercado.

ESQUEMA FIGURADO DO IMPLEMENTO PARA GRAVAÇÕES E CARREGAMENTOS DE FITA K7

ACIMA: COMPONENTES E LIGAÇÕES
ABAIXO: CAIXA PLÁSTICA COM FURAÇÃO PARA INSTALAÇÃO DOS MESMOS

FITA K-7, EXPANSÃO SEM LIMITES PARA O MICROCOMPUTADOR

No capítulo GRAVADOR K-7, A OUTRA METADE DO MICROCOMPUTADOR enfatizamos a utilidade do gravador K-7 como complemento do microcomputador, em razão de o mesmo processar fitas que armazenam programas de microcomputação.

Esse armazenamento constitui em realidade uma verdadeira "memorização" de programas, não havendo nenhuma restrição quanto ao tipo dos mesmos ou seu conteúdo: utilitários, arquivos de dados, aplicativos ou de entretenimento, em linguagem Basic ou de máquina etc.

Assim, constituem as fitas K-7 verdadeira expansão sem limites para o computador, razão por que devem merecer atenção e cuidados adequados, entre os quais destacamos:

- não deixá-las muito próximas do aparelho de TV, ou de alto-falantes de grande potência, ou de outros aparelhos elétricos que geram campos magnéticos de intensidade suficiente para inutilizar gravações feitas nas mesmas;

- não deixá-las expostas ao calor e luz fortes;

- não tocar a fita propriamente dita com os dedos, em razão de a oleosidade natural dos mesmos lhe causar danos;

- mantê-las sempre rebobinadas com a parte transparente da mesma - chamada fita líder - no início, ao melhor abrigo possível do pó e da umidade e, de preferência, com o cartucho em pé, isto é, em posicionamento vertical.

FITAS MAIS INDICADAS

De forma geral, qualquer fita de boa procedência se presta para armazenamento de programas de computação.

Há algumas marcas, todavia, que se recomendam mais, pelo controle rigoroso com que são mantidas suas características e qualidades, podendo-se destacar entre elas BASF e SCOTCH.

As fitas comuns de óxido de ferro, de baixo ruído, servem perfeitamente para a finalidade, não havendo necessidade de se adotar as de custo mais elevado como as de cromo, ferro-cromo ou metal, mesmo porque os gravadores comuns K-7 que são utilizados em computação não são munidos de recursos técnicos capazes de tirar das mesmas os proveitos que suas características oferecem.

Apenas na modalidade de processamento HIGH-SPEED e com um gravador eventualmente mais apropriado, se recomendaria o uso de fita detentora de características melhores de nível de ruído e de gravação de médias e altas frequências.

Atualmente são disponíveis no mercado fitas SCOTCH com duração de 20 minutos, produzidas especialmente para computação. São anunciadas como previamente testadas para evitar o que se chama de "dropout", defeito de gravação que inutiliza um programa e que é causado por falha existente na fita propriamente ou em sua pigmentação.

Com relação a tempo de duração de processamento, as fitas de melhor praticabilidade são as de 20 e 46 minutos, para gravações duplas de um ou dois programas de cada lado.

Quando razões econômicas ou de espaço justificam a utilização de fitas mais longas, recomenda-se que não sejam utilizadas as de duração superior a 60 minutos, por duas razões principais:

- as dificuldades para localização dos programas gravados se tornam maiores e por vezes extremamente cansativas, especialmente quando o gravador utilizado não dispõe de conta-giros ou não foram feitas anotações adequadas no cartucho;

- as probabilidades de surgirem problemas no arrasto da fita durante gravações e reproduções e em rebobinamentos para a frente e para trás aumentam, principalmente quando o rolo pressor de borracha e outras partes do mecanismo do gravador para movimentação da fita começam a apresentar algum desgaste ou folgas excedentes às especificações.

Algumas recomendações para que a manipulação e processamento de uma fita com vários programas não apresente dificuldades:

- usar fitas cujo cartucho plástico seja da melhor transparência e tenha uma escala bem nítida de visualização do ponto de bobinamento. Caso a escala não seja muito nítida, aconselha-se colar no local uma pequena tira de etiqueta adesiva clara, com marcações apropriadas e nítidas, feitas previamente. Evidentemente, se o gravador utilizado dispuser de conta-giros de fita e forem anotados os pontos iniciais de cada gravação feita na fita, a presente recomendação não é válida. Não há dúvida, todavia, que a grande maioria dos aparelhos existentes atualmente em nosso mercado não conta com esse recurso;

- não ter mais do que dois programas longos ou três médios - gravados duplamente e em sequência - em cada lado da fita;

- procurar agrupar numa mesma fita programas de frequência de uso próxima, a fim de que o estiramento e rebobinamento da fita se processe por igual;

- iniciar a gravação de cada programa sempre numa das marcas da escala de visualização do ponto de bobinamento da fita, de maneira que sejam facilitadas as anotações na tampa ou no cartucho da fita e a própria localização do programa quando for efetuar seu carregamento para o computador;

- manter os cabeçotes e o mecanismo do gravador em condições sempre adequadas para o seu melhor desempenho, a fim de que a qualidade de gravação e reprodução de fitas seja otimizada.

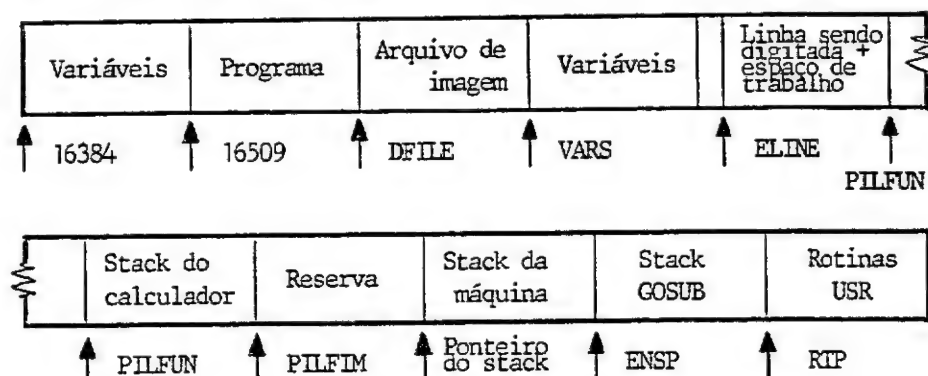
ORGANIZAÇÃO DA MEMÓRIA DO TK 85 E AS VARIÁVEIS DO SISTEMA

Para que funcione dentro dos parâmetros inteligentemente estabelecidos, prestando os excelentes serviços que nos presta, o microcomputador tem sua estrutura "intelectual" dividida em duas memórias: ROM, em inglês "read only memory", e RAM, em inglês "random access memory". Ambas fazem parte do microprocessador ZILOG Z 80, constituído de 65535 "bytes". Byte é uma unidade de 8 bits (bit = unidade binária). Os "bytes" armazenam eletronicamente códigos ou números binários, que indicam ao microprocessador, através de seus milhares de circuitos eletrônicos integrados, quais as ações, funções ou operações que devem ser desempenhadas ou executadas, a fim de que a computação propriamente dita se realize. Os "bytes" também são referidos como "endereços", numerados de 0 a 65535.

A ROM, como diz o nome inglês, só pode ser lida, o que quer dizer que podemos conhecê-la, saber o que contém, estudar o seu funcionamento e utilizá-la, mas é permanente e imutável. Desperta e dorme com o computador, sempre do mesmo jeito, quando este é ligado e desligado. Nela estão contidas todas as rotinas e procedimentos de funcionamento, fornecimento de dados para o monitor de vídeo e para a impressora e o programa chamado interpretador, que traduz as instruções da linguagem BASIC de forma que cada uma acione com precisão a rotina específica de cada função requerida. No TK 85 a ROM ocupa os endereços de memória de 0 a 10240.

A RAM, como também diz o nome em inglês, é a parte da memória à qual temos acesso, isto é, é o local onde colocamos e manipulamos nossos programas, quer em linguagem BASIC ou de máquina, e onde são colocadas as chamadas "variáveis do sistema", cujas funções nos interessam mais de perto e que logo examinaremos.

A RAM ocupa a área de memória dos endereços 16384 a 65535. É subdividida em diversas áreas, cada qual com uma finalidade especificada. Os endereços dos limites iniciais dessas áreas são controlados pelas variáveis do sistema, que fornecem informações precisas de sua localização ao longo de toda a RAM, o que é de muita importância em razão de que algumas dessas áreas são elásticas, isto é, aumentam ou diminuem em função da quantidade de dados que contêm a cada momento, empurrando ou puxando as outras concomitantemente, o que faz com que os endereços ocupados pelas mesmas mudem instantemente. Aprender a conhecer e saber como localizar esses endereços e manipulá-los, em alguns casos alterando os códigos de instruções que contêm, é de grande valia para dominar melhor o microcomputador.

Apresentamos a seguir um esquema figurado da RAM, extraído do manual do usuário do aparelho. O mesmo mostra as diversas áreas mencionadas, bem como sua localização relativa e o nome das variáveis do sistema que controlam e registram os endereços iniciais de cada uma.

ESQUEMA FIGURADO DA RAM

O estudo completo e minucioso das variáveis do sistema e do próprio sistema constitui capítulo à parte e foge do escopo deste trabalho. Deve ser feito com base no manual de especificações e uso do aparelho e em obras especializadas, a partir do momento em que o usuário se sinta habilitado e decida explorar ao máximo toda a potencialidade de seu microcomputador.

Examinaremos aqui apenas algumas dessas variáveis do sistema, as quais não devem ser confundidas com as variáveis comuns numéricas e alfanuméricas utilizadas nos programas vistos anteriormente. Examinaremos apenas as de utilização praticamente obrigatória mesmo por principiantes, com o fito de dar uma idéia de seu desempenho e de como podem ser manipuladas, isto é, "instruídas" e usadas para certas finalidades.

Todas as variáveis do sistema ocupam endereço permanente na RAM e, dependendo de sua função, ocupam um ou dois endereços ou "bytes" no computador.

Vejamos as seguintes:

VARIAVEL	ENDEREÇO(S)	FUNÇÃO
RTP	16388-16389	Indica o endereço inicial da área de memória não utilizável no computador.
LPC	16394-16395	Indica o número da última linha de um programa que esteve em processamento.
DFILE	16396-16397	Indica o começo da área de arquivo dos códigos para imagem de TV.
VARS	16400-16401	Indica o começo da área de variáveis para programação normal.
ELINE	16404-16405	Controla espaços para linha em digitação.
DFSZ	16418	Reserva duas linhas no extremo inferior do vídeo para edição ou digitação.
PXLN	16425-16426	Indica o número do primeiro endereço livre depois de um programa.

Comecemos por DFILE:

Digite: PRINT PEEK 16396+256*PEEK 16397-16509

Tecle NEW LINE. Observe o número que aparece no canto superior esquerdo do vídeo. Deve ser 0.

Digite: 10 PRINT "CANECA" - Repita o comando feito antes.

Tecle NEW LINE. Observe que o número que aparece no vídeo agora é 14. E a quantidade de "bytes" que a linha de programa 10 PRINT "CANECA" ocupa na área de programas na RAM.

Note no esquema figurado da RAM que a área de programas começa no endereço ou "byte" 16509.

O comando PRINT PEEK 16396+256*PEEK 16397-16509 verifica em que endereço se encontra o marco divisório da variável DFILE, que é a primeira depois da área de programas, e subtrai do número achado o do endereço de início da área, 16509, sendo a diferença a quantidade de "bytes" em uso na área de programas.

Assim, para se saber a quantidade de "bytes" de qualquer programa, basta valer-se desse recurso. E quase uma constante a necessidade de ter-se essa informação, para diversas finalidades.

Digite agora o comando como linha de programa, acrescentando-o à linha 10 já existente:

20 PRINT PEEK 16396+256*PEEK 16397-16509

Tecle RUN e NEW LINE. Observe que, além da palavra CANECA, agora o número que aparece no vídeo é 67, correpondendo a 14 "bytes" da linha 10 mais 53 "bytes" da linha 20, que foi incluída como linha de programa e que, portanto, ocupa espaço na memória RAM.

Caso seja necessário verificar apenas o endereço em que se encontra o início da área, basta digitar o comando sem indicar a subtração do endereço 16509: PRINT PEEK 16396+256*PEEK 16397.

Tecle NEW e NEW LINE.

Vejamos agora a variável VARS:

Digite: PRINT PEEK 16400+256*PEEK 16401-16509

Tecle NEW LINE. Observe que aparece agora no vídeo o número 793, embora não haja nenhum programa no computador.

Observando o esquema da RAM, notamos que a variável VARS controla o limite inicial da área de variáveis de programas - não confundir com as variáveis do sistema - marcando portanto o fim da área de arquivo de imagem, isto é, o local onde o microcomputador coloca os códigos de um programa existente na RAM, referentes à imagens de vídeo que de-

vem ser geradas do mesmo. E esta área ocupa permanentemente 793 endereços ou "bytes" da RAM, no caso do TK 85, diferindo do TK 82 C ou TK 83.

Todo programa introduzido no computador tem seus códigos automaticamente copiados e registrados nessa área, de acordo com as etapas do programa, a fim de que sejam geradas as imagens de cada uma na sequência natural.

Digite: 10 PRINT "CANECA"

Tecle NEW LINE.

Digite: PRINT PEEK 16400+256*PEEK 16401-16509

Tecle NEW LINE. Observe o número que aparece agora no vídeo: 807, correspondendo a 14 "bytes" da área de programas e 793 da área de arquivo de imagens.

Se for introduzido um programa que ocupe 5000 "bytes", por exemplo, o número a ser apresentado será 5793.

Tecle NEW e NEW LINE.

A variável ELINE:

Ela registra sempre o endereço inicial da área reservada na RAM para a linha de programa que está sendo digitada e que não passou ainda para a área de programas. Seu marco divisório está logo após a área de variáveis de programas. Assim, podemos também utilizá-la para saber a quantidade total de "bytes" ocupados na RAM por programas que utilizam variáveis, o que é muito importante quando são usadas variáveis extensamente dimensionadas como DIM A$(200,20), DIM A(200), DIM B(200) e outras, por exemplo.

Digite: PRINT PEEK 16404+256*PEEK 16405-16509

Tecle NEW LINE. Observe o número que aparece no vídeo: 794, sendo 793 da área de arquivo de imagens e 1 de fronteira.

Digite agora: DIM A$(200)

Tecle NEW LINE.

Digite de novo: PRINT PEEK 16404+256*PEEK 16405-16509

Tecle NEW LINE. Observe agora o número correspondente à quantidade de "bytes" utilizados pelo dimensionamento da variável A$, somados aos "bytes" da área de arquivo de imagens.

Antes de prosseguirmos, vamos entender melhor o comando já utilizado diversas vezes nas verificações feitas: PRINT PEEK.

PRINT PEEK 16396 significa: imprima (ou escreva) no vídeo o conteúdo ou código do endereço 16396.

PEEK 16397 significa: conteúdo ou código do endereço 16397.

PRINT PEEK 16396+256*PEEK 16397 significa: imprima o resultado da adição do código constante no endereço 16396 com o código que consta no endereço 16397 multiplicado por 256.

Supondo que no momento da verificação os números dos códigos sejam: 139 no endereço 16396 e 64 no endereço 16397, o comando exemplificado por último corresponde a 139+(256*64) ou a 16523, que é o endereço onde se encontra o marco da variável DFILE se no momento houver no computador uma linha de programa 10 PRINT "CANECA".

Sabemos que cada "byte" é uma unidade de 8 bits que armazena um código ou um número binário. A capacidade máxima de um "byte" é armazenar codificação binária correspondente ao número decimal 255. Vimos também que o microprocessador utilizado no microcomputador dispõe de 65535 "bytes", embora essa quantidade seja usada somente nos aparelhos de memória RAM expandida para 48 K "bytes". Por essa razão, a manipulação dos números dos endereços ou dos "bytes" é feita necessaria e inevitavelmente por dois "bytes", utilizando-se o primeiro para a parcela menos significativa do número (a de menor peso ponderado) e o segundo para a parcela mais significativa.

Essa peculiaridade poderá ser melhor compreendida com o exame da próxima variável do sistema:

A variável RTP, quase sempre referida como RAMTOP.

Ela registra o início da área não utilizável na RAM, a área de de topo, mas pode ser manipulada para desempenhar outra função.

Digite: PRINT PEEK 16388+256*PEEK 16389

Tecle NEW LINE. Observe o número 32768 que aparece no vídeo. E o endereço do início da área referida, se o computador tiver 16 K de memória RAM.

Como vimos no esquema, a RAM começa no endereço 16384. Lembremos que um K "byte" corresponde a 1024 "bytes". Portanto, 32768 (= 16384 + 16384) é o endereço do primeiro "byte" não utilizável do sistema.

Digite: PRINT PEEK 16388

Tecle NEW LINE e anote o conteúdo do endereço 16388, que deverá estar impresso no vídeo. Deve ser 0.

Digite: PRINT PEEK 16389

Tecle NEW LINE. Anote também o conteúdo do endereço 16389. Deve ser 128.

De acordo com o que vimos pouco antes sobre a manipulação dos números dos endereços por dois "bytes", temos: 0 (do 1º "byte") + (256*128) (do 2º "byte") = 32768.

A variável RTP pode ser manipulada para baixar seu marco divisório, por exemplo para o endereço 32000, a fim de que sejam colocados acima desse endereço programas ou rotinas que fiquem a salvo dos comandos NEW e CLEAR, o que pode ser um recurso muito valioso em certas circunstâncias. Deve ser levado em conta, todavia, o fato de que também os comandos SAVE e LOAD não tem efeito sobre essa área.

Suponha-se que se queira rodar vários programas que utilizam um mesmo arquivo ou conjunto de dados. Coloca-se o mesmo nessa área, isto é, entre os endereços 32000 e 32768, em linguagem de máquina, carrega-se o programa que vai utilizar esses dados, o carregamento sendo feito normalmente através de LOAD ou de digitação, e faz-se a introdução dos dados do arquivo, através de chamada apropriada do mesmo pelo comando USR, grava-se esse programa em fita K-7, procede-se a um NEW, sem afetar o arquivo na área reservada, carrega-se novo programa para processamento, e assim sucessivamente.

COMO BAIXAR A RTP. Para o endereço 32000, por exemplo:

Digite: POKE 16388,0 e tecle NEW LINE

Digite: POKE 16389,125 e tecle NEW LINE

Tecle NEW e NEW LINE.

Verifique agora onde se encontra o marco divisório da RTP:

Digite: PRINT PEEK 16388+256*PEEK 16389

Tecle NEW LINE e note que será exibido no vídeo o número 32000, que é o endereço onde se encontra agora o marco da RTP.

A título de exercício, baixe a RTP para 29500, lembrando que dividindo-se 29500 por 256 teremos um quociente de 115 com resto 60, equivalendo a 60+(115*256). Se tiver dúvidas sobre a conversão, leia o capítulo SISTEMAS DE NUMERAÇÃO E O COMPUTADOR, que segue ao presente.

A variável DFSZ:

Ela controla o número de linhas na parte inferior da tela de TV, reservando sempre e automaticamente duas linhas ali, para digitação ou edição de comandos ou instruções.

Digite: PRINT PEEK 16418

Tecle NEW LINE. Note que o número 2 que aparece no vídeo é o seu conteúdo e se refere às duas linhas mencionadas. Esse número pode ser manipulado.

Digite:

```
10 FOR C=0 TO 23
20 PRINT "LINHA NUMERO ";C
30 NEXT C
```

Tecle RUN e NEW LINE. Note que, depois de rodado, o programa pára na linha 21 com a notação 5/20, indicando falta de espaço para pros-

seguir. Manipule então a variável DFSZ:

Digite: 5 POKE 16418,0

Tecle NEW LINE, RUN e NEW LINE. Note que o programa foi executado integralmente agora, em consequência da introdução da linha 5, que muda o conteúdo da variável de 2 para 0, anulando assim a reserva feita pela mesma. Manipule novamente a variável:

Digite: 5 POKE 16418,10

Tecle NEW LINE, RUN e NEW LINE. Note que agora o programa foi executado somente até a linha 13.

A necessidade de obter as 24 linhas de vídeo, ou apenas 3, 5, 10 etc., varia em função do tipo de programa e da imaginação de quem o elabora, mas não há dúvida, todavia, que a manipulação da variável DFSZ pode ser muito útil, especialmente para elaboração de jogos e gráficos.

A variável LPC:

Ela registra a última linha que esteve em digitação ou alteração e na qual permanece o cursor de edição do programa.

Sua utilização pode ser prática quando fazemos alguma alteração num programa muito longo e esquecemos o número da linha alterada, sendo necessário novo acesso à mesma para retificação ou nova alteração, como exemplo. Em vez de percorrermos o programa todo, fazendo diversos comandos LIST, voltando atrás etc., podemos fazer apenas o comando da função da variável.

Verifique num programa existente na memória do computador qual a última linha que foi trabalhada:

Digite: PRINT PEEK 16394+256*PEEK 16395

Tecle NEW LINE e em seguida confirme a informação obtida.

A variável PXLN:

Ela registra o primeiro "byte" ou endereço livre após um programa existente na RAM. A indicação desse endereço pode ter muita utilidade em várias circunstâncias de uso do computador, sendo a mais simples delas quando se quer conhecer o espaço livre remanescente na RAM durante a digitação de um programa longo ou após o carregamento de um nessa condição. Para obter essa indicação:

Digite: PRINT PEEK 16425+256*PEEK 16426

Tecle NEW LINE. O número fornecido será o do último endereço ocupado pelo programa existente na RAM, acrescido de 1, que é o primeiro "byte" livre após o mesmo.

Comparando-se o número fornecido com o da RTP - 32768 - saberemos de imediato os "bytes" ainda disponíveis, no caso de um aparelho de 16 K "bytes".

Se essa verificação tiver que ser feita com frequência durante a digitação de um programa longo, o comando da mesma poderá ser feito ou colocado numa linha distante do programa, 9990 por exemplo, e toda vez que se queira aquela informação, bastará utilizar o comando GOTO 9990 e teclar NEW LINE, sem prejuízo para o programa em digitação ou verificação e estudo.

A variável PXLN poderá também indicar diretamente a quantidade de "bytes" disponíveis, adotando-se o seguinte comando ou instrução:

PRINT 32768-(PEEK 16425+256*PEEK 16426)

Ao fazer-se tal comando sem que haja linhas de programa ou dimensionamentos de variáveis no computador, esperar-se-á obter no vídeo o número 16384 como resposta, equivalente a 16 K "bytes" da RAM, já que 1 K "byte" é = 1024 "bytes". Mas, efetivamente, o computador fornecerá o número 16259, portanto 125 "bytes" a menos, em virtude de os mesmos serem ocupados permanentemente pelas variáveis do sistema.

Assim, para efeito de saber realmente qual a disponibilidade de "bytes" na RAM, para utilização em programas, devem ser descontados: 125 "bytes" ocupados pelas variáveis do sistema, mais 793 do arquivo de imagem de TV e mais 100, aproximadamente, ocupados pelos chamados "stacks" existentes na RAM, restando em verdade cerca de 15350 endereços ou "bytes".

Examinamos apenas algumas das variáveis existentes no sistema, e o fizemos mais com o fito de dar uma idéia do que são, como funcionam e como podem ser manipuladas. Mas, como dissemos no início deste capítulo, o estudo completo e profundo das mesmas foge à finalidade deste humilde trabalho.

Entre as diversas outras variáveis do sistema, há mais algumas de muita utilidade prática e outras que valem ser conhecidas no mínimo por suas curiosas e peculiares funções, sobretudo para o usuário interessado em tirar o máximo rendimento de seu microcomputador à medida em que vai se aperfeiçoando em programações mais avançadas.

Recomendamos um exame de todas as variáveis do sistema no manual de proprietário do aparelho. Seu amplo conhecimento e domínio, entretanto, e como já afirmamos, depende de estudo minucioso, feito à parte do presente, e com auxílio de literatura específica.

SISTEMAS DE NUMERAÇÃO E O COMPUTADOR

Qualquer usuário de computador - micro, mini ou macro - inevitavelmente se defrontará com problemas relacionados a sistemas de numeração, quer seja para estudo, compreensão e manipulação do próprio computador, quer seja para conversões necessárias em programações, de forma que é sempre útil ter às mãos dados e tabelas que facilitem a rememorização deste assunto, por vezes intrincado.

NUMERAÇÃO DE BASE 10 OU DECIMAL: Utiliza 10 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Sabemos que um algarismo tem tanto mais valor ou peso quanto mais à esquerda estiver numa escala de posições de um número. Assim, o número 1000 vale 1000 vezes 0001 no sistema decimal porque os valores das posições aumentam em potências de 10 para cada casa esquerda na escala. Exemplo:

NÚMERO	N	N	N	N	
POSIÇÃO	4	3	2	1	
PESO/VALOR	10^3	10^2	10^1	10^0	
VALOR RELATIVO	Nx1000 +	Nx100 +	Nx10 +	Nx1	
Assim, o número	1	0	0	0	vale 1000
porque	(1x1000)+	(0x100) +	(0x10) +	(0x1)	= 1000
O número	3	2	4	7	vale 3247
porque	(3x1000)+	(2x100) +	(4x10) +	(7x1)	= 3247

NUMERAÇÃO DE BASE 2 OU BINÁRIA: Utiliza 2 símbolos: 0 e 1.

Tem seus valores aumentados em potências de 2 para cada casa co-

locada à esquerda na escala de posições. Assim, 1000 em sistema binário vale 8 vezes 0001. Exemplo:

NÚMERO	N	N	N	N	
POSIÇÃO	4	3	2	1	
PESO/VALOR	2^3	2^2	2^1	2^0	
VALOR RELATIVO	Nx8 +	Nx4 +	Nx2 +	Nx1	
Assim, o número	1	0	0	0	vale 8
porque	(1x8) +	(0x4) +	(0x2) +	(0x1)	= 8
O número	1	1	1	1	vale 15
porque	(1x8) +	(1x4) +	(1x2) +	(1x1)	= 15

NUMERAÇÃO DE BASE 16 OU HEXADECIMAL: Utiliza 16 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. A é equivalente a 10 em decimal, B = 11, C = 12, D = 13, E = 14, F = 15.

Os números hexadecimais têm seus valores ou pesos aumentados em potências de 16 para cada casa colocada à esquerda na escala de posições dos dígitos. Assim, 1000 em hexadecimal vale 4096 vezes 0001. Exemplo:

NÚMERO	N	N	N	N	
POSIÇÃO	4	3	2	1	
PESO/VALOR	16^3	16^2	16^1	16^0	
VALOR RELATIVO	Nx4096 +	Nx256 +	Nx16 +	Nx1	
Assim, o número	1	0	0	0	vale 4096
porque	(1x4096)+	(0x256) +	(0x16) +	(0x1)	= 4096
O número	3	1	6	5	vale 12465
porque	(3x4096)+	(1x256) +	(6x16) +	(5x1)	= 12645

NUMERAÇÃO DE BASE 256, SISTEMA ORA PROPOSTO: Utiliza 256 símbolos, compreendendo os números hexadecimais de 00 (=0) a FF, ou os decimais, de 0 a 255. Sua potenciação é de 256 para cada posição à esquerda na escala de posições dos números.

Sua representação é feita sempre entre aspas e com valores atribuídos a cada símbolo, separado um do outro por parênteses. Entre os parênteses é colocada a letra H quando os valores são representados p/ números hexadecimais (sempre em grupos de 2 dígitos cada, como é norma em computação), ou a letra D quando os valores são representados pelos números decimais. Exemplificando:

NÚMERO	" N	ou (H)/(D)	N "		
POSIÇÃO	2		1		
PESO/VALOR	256^1		256^0		
VALOR RELATIVO	Nx256		Nx1		
Assim, o número	" 10	(H)	00 "	....	vale 4096
porque	(16x256)	+	(0x1)	....	= 4096
O número	" FF	(H)	FF "	....	vale 65535
porque	(255x256)	+	(255x1)	....	= 65535
O número	" 117	(D)	48 "	....	vale 30000
porque	(117x256)	+	(48x1)	....	= 30000

Os computadores classificados como de 8 bits (= 1 "byte") são manipulados através de números que variam de 0 a cifras muito elevadas. Os que adotam o microprocessador Z 80 da Zilog, caso do TK e outros da linha Sinclair, são dotados de 65535 "bytes" ou endereços, muitos dos quais têm que ser manipulados com certa frequência.

Mas, como sabemos, 8 bits só podem conter e representar números equivalentes a 0 a 255, num total de 256, de forma que os computadores dessa categoria utilizam 2 "bytes" (cada um com 8 bits) para pro

cessamento de números de 0 a 65535.

Os dois "bytes" são usados pelo computador, nessa função, como números numa escala de posições: o primeiro "byte" (o de endereço mais baixo) na primeira posição, o segundo "byte" (o de endereço mais alto) na posição equivalente à esquerda do primeiro, de forma que o primeiro é de potência 0 e o segundo de potência 1. Se fossem usados 3 "bytes" para essa função, para números superiores a 65535, o terceiro "byte" ocuparia, figuradamente, o 3º lugar à esquerda na escala de posições e seria potenciado ao quadrado, isto é, potência de índice 2, em analogia com os demais sistemas de numeração.

Para computadores até a capacidade mencionada, duas posições apenas são suficientes, por cobrirem toda a gama de números de 0 a 65535.

Seguindo as normas vigentes e supondo que tenhamos que mudar a RTP - como foi visto no capítulo ORGANIZAÇÃO DA MEMORIA DO TK 85 E AS VARIAVEIS DO SISTEMA - para o endereço 32150, vamos encontrar a seguinte indicação: "BAIXAR A RTP PARA 32150". Temos então que converter esse número e anotar os resultados para, em seguida, fazer os comandos necessários.

Se a notação ou instrução para o mesmo fim for feita segundo proposto agora, teremos a indicação feita na seguinte forma: BAIXAR A RTP PARA "125(D)150". De acordo com o exposto mais acima, fazemos os comandos diretos seguintes:

POKE 16388,150	e	NEW LINE
POKE 16389,125	e	NEW LINE
NEW	e	NEW LINE

Se quisermos saber a que endereço corresponde a notação basta efetuarmos: 256x125 (do 2º "byte") + 150 (do 1º "byte"), ou, como já temos visto: 150 + (256x125), resultando 32150.

Quando for conveniente fazer as notações em números hexadecimais num programa em linguagem de máquina, por exemplo, basta substituirmos os valores decimais pelos seus correspondentes hexadecimais, com auxílio da tabela de equivalência HEXA-DECI, introduzida logo após o presente capítulo, com a conveniência de não se ter que se preocupar com cálculos para esse fim.

A conversão de um número decimal para um de base 256 faz-se com uma simples operação aritmética de divisão. Seja o número 16509, como exemplo:

16509 dividido por 256 = 64, com resto 125.

Logo, "64(D)125" é o equivalente NUMERO DE BASE 256.

Seja o número 31450, como outro exemplo:

31450 dividido por 256 = 122, com resto 218.

Logo, "122(D)218" é o equivalente NUMERO DE BASE 256.

As vantagens desse sistema, numa análise mais imediata:

- notações diretas e concisas;

- maior facilidade na introdução de dados correlatos no computador, já que sua interpretação é direta e imediata;

- maior rapidez nas conversões necessárias.

Embora não aparente à primeira vista sua grande utilidade, a mesma será facilmente percebida pelo usuário de computador tão logo a necessidade de manipulação de "bytes" ou endereços se manifeste.

TABELA DOS SIMBOLOS DOS SISTEMAS DE NUMERAÇÃO E EQUIVALENCIAS

(Os sublinhados constituem os símbolos de cada sistema.)

DECIMAL:	HEXADECIMAL:	BINÁRIO:
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Em computação é generalizado o uso dos dígitos hexadecimais em grupos de dois, separados ou agrupados. Por exemplo: 00, 09, C5, A7, B306 etc., para facilidade de leitura e compreensão, o mesmo acontecendo com os números binários, que são agrupados em quatro dígitos. Por exemlo: 0001, 0010 0111 etc. A ordem crescente é da direita para a esquerda, em ambos os casos.

TABELA HEXA-DECI - COMBINAÇÕES DOS SIMBOLOS HEXADECIMAIS E SEUS EQUIVALENTES DECIMAIS

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Os 16 símbolos hexadecimais encontram-se na primeira linha superior e na primeira coluna à esquerda, ambos fora do quadro. Dentro do quadro encontram-se os números decimais de 0 a 255, que correspondem, em equivalência, às combinações possíveis dos símbolos hexadecimais.

As combinações são obtidas através do encontro das linhas com as colunas em que se encontram os símbolos. O ponto de encontro das mesmas, dentro do quadro, dá o número decimal equivalente. Exemplos:

Para se determinar o equivalente decimal do nº hexadecimal B7, procura-se na 1ª coluna à esquerda, fora do quadro, a letra B e percorre-se sua linha até a coluna em que se encontra o nº 7, na 1ª linha fora do quadro. No ponto de encontro das mesmas encontra-se o número decimal equivalente, no caso o nº 183.

Para se determinar o equivalente hexadecimal do nº decimal 166, procura-se este no quadro dos números decimais e determina-se a linha em que se encontra o mesmo em relação à 1ª coluna à esquerda, fora do quadro, achando-se A. Em seguida, determina-se a coluna em que se encontra o mesmo em relação à 1ª linha superior, fora do quadro, achando-se o símbolo 6. O nº hexadecimal será A6, equivalente a 166 decimal.

CONVERSÃO DECIMAL PARA BINARIO (BASE 10 PARA BASE 2)

A conversão é feita através de divisões sucessivas do número decimal por 2, anotando-se os restos das divisões, até o quociente tornar-se menor do que o divisor 2. O número convertido é formado pelo último quociente e os restos das divisões efetuadas, lidos na orden inversa àquela em que foram obtidos.

Nº DECIMAL		Nº BINARIO	PROCEDIMENTO
8	=	1000	8 : 2 = 4 ... com resto 0 4 : 2 = 2 ... com resto 0 2 : 2 = 1 ... com resto 0
114	=	111 0010	114 : 2 = 57 ... com resto 0 57 : 2 = 28 ... com resto 1 28 : 2 = 14 ... com resto 0 14 : 2 = 7 ... com resto 0 7 : 2 = 3 ... com resto 1 3 : 2 = 1 ... com resto 1
1520	=	101 1111 0000	1520 : 2 = 760 ... com resto 0 760 : 2 = 380 ... com resto 0 380 : 2 = 190 ... com resto 0 190 : 2 = 95 ... com resto 0 95 : 2 = 47 ... com resto 1 47 : 2 = 23 ... com resto 1 23 : 2 = 11 ... com resto 1 11 : 2 = 5 ... com resto 1 5 : 2 = 2 ... com resto 1 2 : 2 = 1 ... com resto 0

Observação: Quando o número decimal a ser convertido for muito grande, a conversão se torna mais rápida se for feita primeiramente para números de base 256, em seguida convertidos para hexadecimais e finalmente para binários.

CONVERSÃO BINARIO PARA DECIMAL (BASE 2 PARA BASE 10)

A conversão é feita multiplicando-se os dígitos binários pelo valor relativo à potenciação da base 2 nas posições correspondentes aos mesmos na escala de pesos, sendo a soma desses produtos o número convertido. O uso da tabela que se segue facilitará as conversões:

TABELA DE POTENCIAS DAS POSIÇÕES E VALORES RELATIVOS DAS MESMAS

POSIÇÃO	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
POTENCIA	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
VALOR RELATIVO	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Exemplos:

Nº BINARIO		Nº DECIMAL	PROCEDIMENTO
1000	=	8	1 x 8 = 8 (8 é a potenciação da base 2 na posição 4).
11 1000	=	56	32 + 16 + 8 - Soma dos valores relativos às posições 6, 5 e 4, onde o dígito é 1.
1010 1111	=	175	128 + 32 + 8 + 4 + 2 + 1 - Soma dos produtos das posições em que o dígito é 1.
11 0101 1001	=	857	512 + 256 + 64 + 16 + 8 + 1 - Idem.
1100 0011 0101	=	3125	2048 + 1024 + 32 + 16 + 4 + 1 - Idem.

CONVERSÃO DECIMAL PARA HEXADECIMAL (BASE 10 PARA BASE 16)

A conversão é feita através de divisões sucessivas do número decimal por 16, anotando-se os restos das divisões, até o quociente tornar-se menor do que o divisor 16. O número convertido é formado pelo último quociente e os restos das divisões feitas, lidos na ordem inversa àquela em que foram obtidos, e convertidos para os símbolos hexadecimais correspondentes. Seguem-se exemplos:

Nº DECIMAL		Nº HEXADECIMAL	PROCEDIMENTO
114	=	72	114 : 16 = 7 com resto 2
236	=	EC (E=14/C=12)	236 : 16 = 14 com resto 12
1520	=	5F0 (F=15)	1520 : 16 = 95 com resto 0 95 : 16 = 5 com resto 15
16384	=	4000	16384 : 16 = 1024 com resto 0 1024 : 16 = 64 com resto 0 64 : 16 = 4 com resto 0
24564	=	5FF5 (F=15)	24565 : 16 = 1535 com resto 5 1535 : 16 = 95 com resto 15 95 : 16 = 5 com resto 15

Observação: A conversão de números decimais até 255 pode ser feita direta e facilmente com auxílio da tabela HEXA-DECI.

A conversão de números decimais superiores a 255 pode ser feita primeiramente para números de base 256 e estes convertidos diretamente com auxílio da mesma tabela HEXA-DECI. Exemplos:

Nº DECIMAL		Nº HEXADECIMAL	PROCEDIMENTO
32600	=	7F58	32600 : 256 = 127 com resto 88 (Tabela: 127 = 7F e 88 = 58).
30150	=	75C6	30150 : 256 = 117 com resto 198 (Tabela: 117 = 75 e 198 = C6).

CONVERSÃO HEXADECIMAL PARA DECIMAL (BASE 16 PARA BASE 10)

A conversão é feita: primeiramente convertendo-se os símbolos hexadecimais para seus equivalentes decimais e em seguida multiplicando-se estes pelo valor relativo à potenciação da base 16 nas posições correspondentes aos mesmos na escala de pesos, sendo a soma desses produtos o número convertido.

São apresentadas a seguir duas tabelas que facilitam conversões:

TABELA DE POTENCIAS DAS POSIÇÕES (BASE 16)

POSIÇÃO	4	3	2	1
POTENCIA	16^3	16^2	16^1	16^0
VALOR RELATIVO	* 4096	* 256	* 16	* 1

TABELA DE EQUIVALENCIA HEXADECIMAL-DECIMAL

0 = 0	4 = 4	8 = 8	C = 12
1 = 1	5 = 5	9 = 9	D = 13
2 = 2	6 = 6	A = 10	E = 14
3 = 3	7 = 7	B = 11	F = 15

Exemplos:

Nº HEXADECIMAL		Nº DECIMAL	PROCEDIMENTO
1F	=	31	(1x16)+(15x1)
93	=	147	(9x16)+(3x1)
4E4	=	1252	(4x256)+(14x16)+(4x1)
4000	=	16384	(4x4096)+(0x256)+(0x16)+(0x1)
7F58	=	32600	(7x4096)+(15x256)+(5x16)+(8x1)

4321 = Posições

Obs.: Para conversões de números hexadecimais de dois símbolos apenas, a conversão poderá ser feita diretamente com auxílio da tabela

<u>CONVERSÕES RAPIDAS:</u> Números hexadecimais de 3 ou 4 dígitos poderão ser considerados em base 256, convertidos para números de base 256 em decimal, com auxílio da tabela HEXA-DECI, e em seguida para números decimais.

Tomemos como exemplo o número 7F58:

- considerado em base 256 = "7F(H)58)"

Consultando a tabela HEXA-DECI, acharemos: 7F = 127, 58 = 88.

- convertendo-se para base 256 decimal = "127(D)88"
- resultando: (127x256)+88 = 32600.

Outro exemplo: número hexadecimal 63D8:

- considerado em base 256 = "63(H)D8"

Consultando a tabela HEXA-DECI, acharemos: 63 = 99, D8 = 216.

- convertendo-se para base 256 decimal = "99(D)216)"
- resultando: (99x256)+216 = 25560.

CONVERSÃO BINARIO PARA HEXADECIMAL (BASE 2 PARA BASE 16)

CONVERSÃO HEXADECIMAL PARA BINARIO (BASE 16 PARA BASE 2)

Os números binários são escritos em grupos de quatro dígitos, separados uns dos outros por espaço, sendo cada grupo convertido para o correspondente hexadecimal, de acordo com sua equivalência:

A conversão de números hexadecimais para binários é feita pelo processo inverso.

E apresentada a seguir uma tabela de equivalência, para facilitar as conversões:

TABELA DE EQUIVALENCIA NUMEROS HEXADECIMAIS E BINARIOS

0	0000	4	0100	8	1000	C	1100
1	0001	5	0101	9	1001	D	1101
2	0010	6	0110	A	1010	E	1110
3	0011	7	0111	B	1011	F	1111

Seguem-se exemplos de conversões:

Nº BINARIO	...	1010	1001	0010	0001	
	=	A	9	2	1	... Nº HEXADECIMAL.
Nº BINARIO	...		0111	1011		
	=		7	B		... Nº HEXADECIMAL.
Nº HEXADECIMAL	...	A	5	3	C	
	=	1010	0101	0011	1100	... Nº BINARIO.
Nº HEXADECIMAL	...		1	F		
	=		0001	1111		... Nº BINARIO.

FUNÇÕES MATEMÁTICAS NO TK

Podemos afirmar, sem exagero, que as funções matemáticas do TK, aliadas aos recursos de programação de que dispõe, transformam-no numa poderosa calculadora cujos limites são apenas os conhecimentos e a imaginação de seu usuário.

Não é nosso propósito explorar nestas poucas páginas as fabulosas possibilidades que esse maravilhoso aparelho oferece, nem contamos com capacidade para tanto.

Atendo-nos ao escopo deste trabalho, cumpre-nos apenas relacionar e descrever sucintamente as funções disponíveis no mesmo e, através de pequenos programas, propiciar ao usuário principiante noções básicas de sua utilização, deixando a seu cargo a exploração adequada e máxima desses recursos para tão logo quanto domínio e conhecimentos seus lhe permitirem.

FUNÇÃO ABS: Através desta função o computador fornece o valor absoluto ou "módulo" de um valor numérico relativo. "Módulo" ou valor absoluto de -347 é 347, de + ou -1000 é 1000.

FUNÇÃO SGN: Fornece o valor unitário +1 ou -1 (sinal) gerador de um número relativo positivo ou negativo, ou o fator 0 de um número neutro.

PARÊNTESES (): Cabe observar aqui que os parênteses, quando usados em expressões matemáticas, funcionam como comandos especiais, indicando ao computador que as instruções de operações contidas pelos mesmos devem ser executadas prioritariamente em relação às demais.

A seguir é apresentado um pequeno programa que ilustra as duas funções acima descritas, ABS e SGN, além de exemplificar alguns outros usos das funções INT, RND, IF, já vistas no capítulo O BASIC SEM RODEIOS, ATRAVÉS DO USO.

Tendo em vista a necessidade de exatidão e utilização de certos recursos gráficos disponíveis somente no próprio microcomputador TK, a partir deste ponto a maioria dos programas que serão introduzidos neste trabalho será apresentada na forma de listagens impressas pela engenhosa impressora produzida especialmente para microcomputadores da linha Sinclair.

Considerando a finalidade didática deste trabalho, as listagens, não obstante não serem introduzidas precedidas da palavra "digite", deverão ser normalmente digitadas pelo leitor praticante, já que é esta a melhor maneira de fixar os conhecimentos adquiridos.

```
  1 REM FUNCOES ABS E SGN
  5 LET A=INT (RND*10)
 10 IF A>=5 THEN LET A=A*-1
 20 PRINT "A = ";A
 30 LET B=-INT (RND*10)
 40 PRINT "B = ";B
 50 PRINT ,,"MULTIPLICACAO"
 60 LET C=A*B
 70 PRINT "A*B=C; C = ";C
 80 PRINT "SINAL DE C = ";SGN C
 90 PRINT "VALOR ABSOLUTO DE C
= ";ABS C
100 PRINT ,,,,TAB 4;" - + - + -
+ - "
110 PRINT ,,"ADICAO"
120 LET D=A+B
130 PRINT "A+B=D; D = ";D
140 PRINT "SINAL DE D = ";SGN D
150 PRINT "VALOR ABSOLUTO DE D
= ";ABS D
160 PRINT ,,,"APERTE UMA TECLA"
170 PAUSE 4E4
180 CLS
190 GOTO 5
```

Tecle RUN e NEW LINE e observe como são interpretados os números gerados pelas instruções das linhas 5, 10 e 30, com sinais positivos e negativos. As linhas 80, 90 e 140 ilustram as funções SGN e ABS, inclusive a linha 150. Rode o programa algumas vezes para melhor assimilar seu desempenho. Note como a linha 10 torna negativo o número gerado pela linha 5, caso o mesmo seja igual ou exceda o valor 5.

O resultado típico de uma rodada do programa é apresentado a se-

guir, copiado pela impressora diretamente do vídeo:

```
A = -7
B = -3

MULTIPLICACAO
A*B=C; C = 21
SINAL DE C = 1
VALOR ABSOLUTO DE C = 21

      - - - - - - -

ADICAO
A+B=D; D = -10
SINAL DE D = -1
VALOR ABSOLUTO DE D = 10
                        APERTE UMA TECLA
```

<u>FUNÇÃO EXP:</u> Fornece o número "e" elevado à potência indicada. O número "e", cujo valor é 2,71828..., é a base dos logarítmos naturais ou neperianos (de John Neper), assim como 10 é a base dos logarítmos decimais ou de Briggs.

Sabemos que o logarítmo de um número é o expoente a que se deve elevar a base adotada, isto é, a base do sistema de numeração ao qual pertence esse número, para obtê-lo, e que a base de um sistema de logarítmos é o número cujo logarítmo nesse sistema é a unidade.

Assim, 1 é o logarítmo natural de "e" (ou 2,71828), como 1 é o logarítmo decimal de 10, sendo $\text{"e"}^1 = 2{,}71828$ e $10^1 = 10$.

Elevar "e" a uma potência indicada, por exemplo "e"^3, significa elevar 2,71828 à sua terceira potência, resultando 20,08553. Elevar 10 à sua terceira potência resulta 1000.

Os computadores de lógica SINCLAIR incluem o sistema neperiano em suas funções já definidas como rotinas internas. Isto quer significar que, desejando-se elevar "e" à 3ª potência, basta comandar PRINT EXP 3. O resultado será 20.085537. (Lembremos que no computador a vírgula é substituída pelo ponto, em se tratando de números.)

A potenciação de outros números no TK é feita através da função simbolizada por ** (2 asteriscos, na tecla H). Para elevar 10 à terceira potência, por exemplo, deverá ser feito o comando PRINT 10**3. O resultado, obviamente, será 1000.

Experimente comandar:

```
PRINT EXP 3
PRINT 10**3
PRINT EXP 12
PRINT 10**12
```

<u>FUNÇÃO LN:</u> Fornece um logarítmo "natural", isto é, da base "e". LN é a função inversa de EXP.

Experimente comandar:

```
PRINT LN 20.085537
```

O resultado será 3, obviamente, que é o logarítmo natural do número fornecido.

O TK não inclui em suas funções já definidas internamente o fornecimento de logarítmos decimais. Para obtê-los, é necessário dividir o LN do número em operação por LN 10. Assim, para obter o logarítmo decimal de 1000 comande:

PRINT LN 1000/LN 10, resultando 3.

Experimente ainda:

```
PRINT LN 15625/LN 10
```

<u>PI</u> (Letra grega π): Não obstante estar incluído como função matemática, na verdade representa apenas o valor numérico da relação constante entre o comprimento de uma circunferência e o comprimento de seu raio, ambos expressos com uma mesma unidade. O valor aproximado de PI é 3,141592653...

Um exemplo clássico da aplicação de PI é a medida da circunferência:

COMPRIMENTO DA CIRCUNFERENCIA = Rx2PI ou DxPI, onde R = raio e D = diâmetro.

Experimente o comando direto:

PRINT 0,7xPI

onde 0,7 equivale a 0,7 m e representa o diâmetro da roda de uma bicicleta, e constate que a cada volta dada a roda faz um percurso de aproximadamente 2,2 m.

FUNÇÃO SQR: Fornece a raiz quadrada de um número X.

Rememorando, a raiz de um número é a quantidade que, tomada determinadas vezes como fator, forma outra quantidade chamada potência. O número de vezes que a raiz entra como fator é chamado índice. A raiz é indicada pelo símbolo $\sqrt{\ }$, chamado radical.

A resolução da expressão $\sqrt{144}$ é obtida no microcomputador através do comando:

PRINT SQR 144

cujo resultado será 12, obviamente.

Extração de raiz quadrada ou de índice 2 é uma função já definida no microprocessador do TK, não havendo no mesmo, entretanto, definição para extração de raízes de outros índices, como cúbico, quarto, quinto etc.

No TK a extração de raízes de índice superior a 2 é feita utilizando-se o recurso inverso, ou seja: elevar o número à potência do valor inverso do índice da raiz, já que, como sabemos, extrair a raiz de um número equivale a elevar esse número à potência $\frac{1}{X}$, onde X é o índice da raiz que se quer extrair. Assim:

$\sqrt[5]{3125}$ equivale a $(3125)^{\frac{1}{5}}$, cuja raiz é 5.

$\sqrt[5]{759375}$ equivale a $(759375)^{\frac{1}{5}}$, cuja raiz é 15.

FUNÇÕES SIN, COS, TAN:

SIN fornece o valor do seno de um ângulo no círculo, sendo o círculo considerado o trigonométrico.

COS idem do co-seno.

TAN idem da tangente.

O ângulo cujos valores SIN, COS e TAN são desejados deve ser expresso em radianos, sendo um radiano = 0,01745329 grau. Assim, um ângulo de 30 graus é = a 0,5235987 radiano. Para se obter um dos valores em questão de um ângulo de 30 graus, por exemplo, deve ser feito o seguinte comando

PRINT SIN 0.5235987

O resultado obtido, como sabemos, será um valor relativo - realmente relativo - ao raio do círculo trigonométrico em que está inscrito o triângulo a que pertence o ângulo.

FUNÇÕES ASN, ACN, ATN:

ASN fornece a medida em radianos do arco-seno.

ACN idem do arco-co-seno.

ATN idem do arco-tangente.

Estas medidas são calculadas pelo computador com base nos valores respectivos de SIN, COS e TAN que lhe são fornecidos, já que são funções inversas umas das outras.

Assim, se se quiser a medida do arco de um seno de valor 0,5, o comando a ser feito é:

PRINT ASN 0.5

cujo resultado será 0.5235987 radiano (= 30 graus), que é o ângulo ou arco ao qual pertence o seno.

Segue-se um pequeno programa para ilustrar o que foi explanado:

```
  1 REM FUNCOES TRIGONOMETRICAS
  5 PRINT "ANGULO EM GRAUS? ";
 10 INPUT A
 20 PRINT A
 30 IF A>=180 THEN GOTO 160
 40 LET B=A*0.017453293
 50 PRINT ,," = ANGULO DE ";B;"
 RADIANOS"
 60 PRINT ,,"SENO = ";SIN B
 70 PRINT "CO-SENO = ";COS B
 80 PRINT "TANGENTE = ";TAN B
 90 PRINT ,,"ARCO-SENO = ";ASN
(SIN B)
 100 PRINT "ARCO-CO-SENO = ";ACS
 (COS B)
 110 PRINT "ARCO-TANGENTE = ";AT
N (TAN B)
 120 PRINT ,,"APERTE QUALQUER TE
CLA"
 130 PAUSE 4E4
 140 CLS
 150 GOTO 5
 160 PRINT ,,"CALCULO IMPOSSIVEL
"
 170 PRINT ,,"ENTRE COM ANGULO M
ENOR: ";
 180 GOTO 10
```

O resultado típico de uma rodada do programa é apresentado a seguir, copiado diretamente do vídeo pela impressora:

```
ANGULO EM GRAUS? 33

 = ANGULO DE 0.57595867 RADIANOS

SENO = 0.54463905
CO-SENO = 0.83867056
TANGENTE = 0.64940762

ARCO-SENO = 0.57595867
ARCO-CO-SENO = 0.57595867
ARCO-TANGENTE = 0.57595867

APERTE QUALQUER TECLA
```

Atenção: Para maior facilidade e compreensão, o ângulo cujos valores serão calculados e fornecidos pelo microcomputador deverá ser

expresso em graus. Através da instrução da linha 40 do programa o computador faz a conversão para radianos.

Tecle RUN e NEW LINE. Forneça várias medidas de ângulos para verificar ou estudar os resultados.

Deve ser observado ainda que as funções ACS e ASN fornecem resultados errados se os argumentos, isto é, os valores fornecidos de seno - SIN - e co-seno - COS - não estiverem entre -1 e 1, pois os mesmos não existem abaixo ou acima desses valores, razão pela qual a linha 30 do programa acima limita os cálculos a ângulos de até 179 graus.

Conversões de graus para radianos podem ser feitas através da seguinte fórmula:

G/180xPI, onde G = graus. E de radianos para graus:

R/PIx180, onde R = radianos.

Seguem-se dois programas relacionados com trigonometria, ambos elaborados para exibirem os mais variados gráficos das funções seno e co-seno e cálculos e gráficos da tangente:

```
   1 REM GRAFICOS DO SENO E DO C
O-SENO
  10 FAST
  20 PRINT "COMPRIMENTO? (ATE 32
) ";
  30 INPUT X
  40 PRINT X
  50 PRINT "AMPLITUDE? (ATE 21)
";
  60 INPUT Y
  70 PRINT Y
  80 PRINT ,,"SENO (S) OU CO-SEN
O (C ?"
  90 INPUT A$
 100 IF A$="" THEN GOTO 100
 110 IF A$<>"S" THEN GOTO 180
 120 CLS
 130 FOR I=0 TO 63
 140 PLOT I,Y
 150 PLOT I,Y+Y*SIN (I/X*PI)
 160 NEXT I
 170 GOTO 230
 180 CLS
 190 FOR I=0 TO 63
```

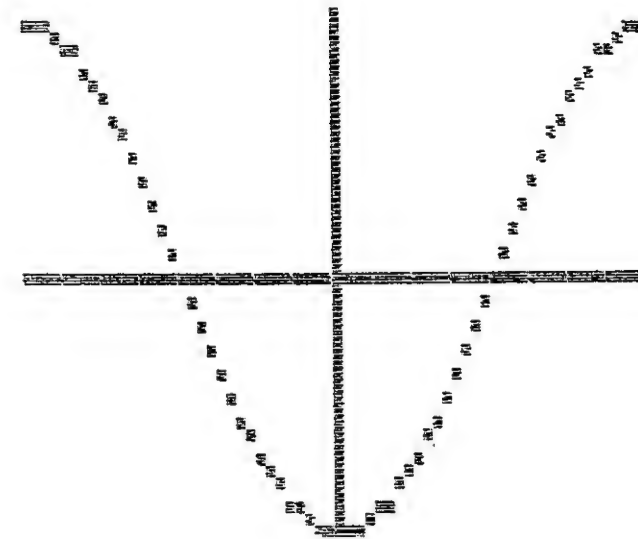
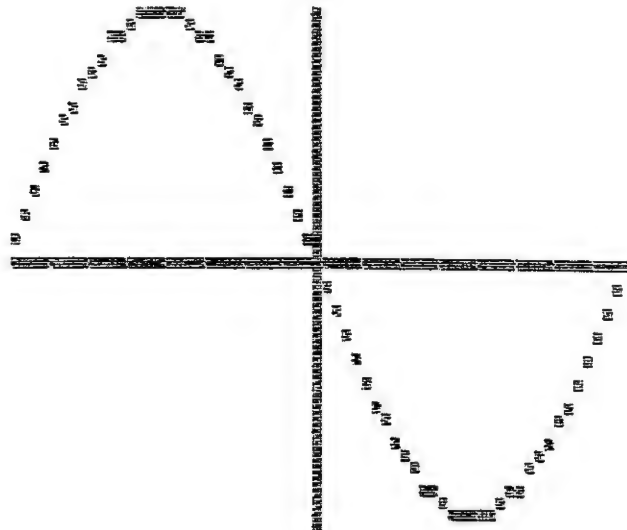
```
200 PLOT I,Y
210 PLOT I,Y+Y*COS (I/X*PI)
220 NEXT I
230 FOR I=0 TO 43
240 PLOT 32,I
250 NEXT I
260 PAUSE 4E4
270 CLS
280 GOTO 10
```

Depois de digitado o programa:

Tecle RUN e NEW LINE. Forneça os dados que o computador solicitar através do vídeo, para que o gráfico correspondente seja exibido.

Aperte qualquer tecla - menos BREAK - para novos ciclos do programa, com a exibição de gráficos diferentes, que serão obtidos mediante o fornecimento de medidas diversas de comprimento e amplitude.

Rode o programa algumas vezes a fim de observar bem os efeitos executados pelas instruções das linhas 10 a 150.

São apresentados a seguir dois gráficos típicos do programa:

```
  1 REM CALCULO E GRAFICO DA TA
NGENTE
 10 PRINT "ANGULO? (ATE 75▪)"
 20 INPUT A
 30 IF A>75 THEN GOTO 20
 40 PRINT ,,"GRAUS:",,A
 50 PRINT ,,"TANG. DE A:"
 60 LET B=A*0.017453293
 70 LET T=TAN B
 80 PRINT T
 90 FOR C=0 TO 63
100 PLOT C,0
110 NEXT C
120 FOR X=0 TO 63
130 IF T*X>44 THEN GOTO 160
140 PLOT X,INT (T*X)
150 NEXT X
160 PRINT AT 20,5;"ANG A"
170 PRINT AT 3,10;"▪";AT 17,20;
"▪"
180 PLOT 41,3
190 PLOT 40,14
200 PLOT 38,20
210 PLOT 34,26
220 PLOT 30,30
230 PLOT 26,34
240 PLOT 14,40
250 PLOT 8,41
260 FOR C=0 TO 43
```

```
270 PLOT 42,C
280 NEXT C
290 PRINT AT 19,23;"PRESSIONE";
AT 20,23;"UMA TECLA"
300 PAUSE 4E4
310 CLS
320 GOTO 10
```

Depois de digitado o programa:

Tecle RUN e NEW LINE. Forneça o dado solicitado pelo computador e, como de costume, observe bem os diversos efeitos do programa.

Experimente entrar com um ângulo de 45 graus e note como a tangente do gráfico corresponde ao cálculo fornecido, isto é: TANGENTE = 1 RAIO.

A linha 30 do programa limita os ângulos a serem introduzidos a 75 graus em razão das limitações do vídeo. As linhas 180 a 260 criam o segmento do círculo. A linha 60 transforma graus em radianos, que é o sistema utilizado pelo microcomputador.

Segue-se um gráfico típico do programa, copiado pela impressora.

PROGRAMAS FINANCEIROS

A capacidade e possibilidades de cálculos de um microcomputador são praticamente ilimitadas.

As funções matemáticas já incluídas no seu sistema operacional e os inumeráveis recursos de programação de sua linguagem de interpretação, inteligente e criativamente conjugados pelo usuário, permitem realizações verdadeiramente incríveis. Quem conhece alguns dos melhores programas utilitários e recreativos disponíveis no mercado sabe quão verdadeira é essa assertiva.

Apresentamos a seguir cinco pequenos programas financeiros, de uso pessoal ou familiar praticamente obrigatórios nos dias atuais, que poderão dar uma pálida idéia das potencialidades descritas, ao mesmo tempo que oferecem exemplos de recursos para programas afins.

Os quatro primeiros programas dispensam instruções para sua utilização. Depois de postos em operação no microcomputador, será apenas necessário introduzir via teclado os dados solicitados através do vídeo para obter prontamente os resultados elaborados pelo computador.

Após a listagem de cada programa é apresentado um quadro típico do mesmo, copiado diretamente do vídeo pela impressora.

```
  5 REM APLICACAO DE CAPITAL
 10 PRINT "CAPITAL APLICADO?"
 20 INPUT CI
 30 PRINT ,,TAB 5;"CR$ ";CI
 40 PRINT ,,"TAXA MENSAL? - %=
 50 INPUT I
 60 PRINT I
 70 PRINT ,,"PRAZO EM MESES? =
 80 INPUT T
 90 PRINT T
100 LET CF=CI*(1+I/100)**T
```

```
 110 PRINT ,,"CAPITAL FINAL: CR$
";CF
 120 PRINT ,,"NOVO CALCULO? (S/N
)"
 130 INPUT R$
 140 IF R$<>"N" THEN GOTO 160
 150 STOP
 160 CLS
 170 GOTO 10
```

```
CAPITAL APLICADO?

     CR$ 14000.525

TAXA MENSAL? - ª/ª = 9.47

PRAZO EM MESES? = 12

CAPITAL FINAL: CR$ 41465.388

NOVO CALCULO? (S/N)
```

```
   5 REM APLICACAO PROGRAMADA
  10 LET CF=0
  20 PRINT "PARCELA MENSAL?"
  30 INPUT P
  40 PRINT ,,TAB 5;"CR$ ";P
  50 PRINT ,,"TAXA MENSAL? - ª/ª
= ";
  60 INPUT I
  70 PRINT I
  80 PRINT ,,"QUANTAS PARCELAS?
= ";
  90 INPUT T
 100 PRINT T
 110 FAST
 120 FOR E=T TO 1 STEP -1
 130 LET CA=P*(1+I/100)**E
 140 LET CF=CF+CA
 150 NEXT E
 160 SLOW
 170 PRINT ,,"CAPITAL FINAL ";T+
1;"ª MES:",,TAB 5;"CR$ ";CF
 180 PRINT ,,"NOVO CALCULO? (S/N
)"
 190 INPUT R$
 200 IF R$<>"N" THEN GOTO 220
 210 STOP
 220 CLS
 230 GOTO 10
```



```
PARCELA MENSAL?

     CR$ 250.255

TAXA MENSAL? - ª/ª = 9.45

QUANTAS PARCELAS? = 12

CAPITAL FINAL 13ª MES:

     CR$ 5667.1069

NOVO CALCULO? (S/N)
```

```
   5 REM TAXA DE FINANCIAMENTO
  10 PRINT "VALOR FINANCIADO?"
  20 INPUT V
  30 PRINT ,,TAB 5;"CR$ ";V
  40 PRINT ,,"VALOR DA PRESTACAO
?"
  50 INPUT P
  60 PRINT ,,TAB 5;"CR$ ";P
  70 PRINT ,,"QUANTAS PRESTACOES
? = ";
  80 INPUT N
  90 PRINT N
 100 PRINT ,,"AGUARDE UNS INSTAN
TES..."
 110 LET I=((P*N/V)**(1/(N+1)))-
1
 120 LET J=(V*I)/(1-(1+I)**(-N))
 130 IF ABS (P-J)>01 THEN GOTO 1
60
 140 PRINT ,,"TAXA DE JUROS (ª/ª
): ";I*100
 150 GOTO 180
 160 LET I=I+((P-J)/P)*I
 170 GOTO 120
 180 PRINT ,,"NOVO CALCULO? (S/N
)"
 190 INPUT R$
 200 IF R$<>"N" THEN GOTO 220
 210 STOP
 220 CLS
 230 GOTO 10
```

NOTA: Para tornar o programa mais rápido:
- Substituir a linha 100 por 100 FAST.
- Alterar o número da linha 180 para 185.
- Acrescentar uma linha: 180 SLOW.

```
VALOR FINANCIADO?

      CR$ 14345.55

VALOR DA PRESTACAO?

      CR$ 2550.234

QUANTAS PRESTACOES? = 12

AGUARDE UNS INSTANTES...

TAXA DE JUROS  (‰): 14.137366

NOVO CALCULO? (S/N)
```

Observação: Os cálculos de juros compostos sobre financiamentos com amortizações mensais são bastante complexos, excedendo os limites da finalidade desta obra, de forma que o programa oferece uma aproximação ideal da realidade, não sendo, todavia, do ponto de vista matemático-financeiro, rigorosamente preciso.

```
  5 REM VALOR PRESTACAO MENSAL
 10 PRINT "VALOR A FINANCIAR?"
 20 INPUT V
 30 PRINT ,,TAB 5;"CR$ ";V
 40 PRINT ,,"TAXA DE JUROS? (‰
) = ";
 50 INPUT I
 60 PRINT I
 70 PRINT ,,"QUANTAS PARCELAS?
= ";
 80 INPUT N
 90 PRINT N
100 LET P=V*((I/100)/(1-(1+I/10
0)**(-N)))
110 PRINT ,,"VALOR DA PRESTACAO
:"
120 PRINT ,,TAB 5;"CR$ ";P
130 PRINT ,,"NOVO CALCULO? (S/N
)"
140 INPUT R$
150 IF R$<>"N" THEN GOTO 170
160 STOP
170 CLS
180 GOTO 10
```



```
VALOR A FINANCIAR?

      CR$ 12583.277

TAXA DE JUROS? (‰) = 14.25

QUANTAS PARCELAS? = 12

VALOR DA PRESTACAO.

      CR$ 2247.5041

NOVO CALCULO? (S/N)
```

```
  5 REM CADERNETA DE POUPANCA
 10 LET N=12
 20 DIM T(N)
 30 PRINT "VALOR APLICADO? - CR
$ ";
 40 INPUT V
 50 GOSUB 330
 60 PRINT AT 0,32-LEN V$;V$
 70 PRINT
 80 FOR C=1 TO N
 90 IF C>=2 THEN GOSUB 250
100 IF C=7 THEN CLS
110 PRINT "COR MON ";C;"° MES?:
";
120 INPUT T(C)
130 IF T(C)>20 THEN GOTO 120
140 PRINT T(C);
150 PRINT " ‰ "
160 IF T(C)=0 THEN GOTO 440
170 PRINT "CAPITAL+COR+JUROS= "
;
180 GOSUB 220
190 PRINT TAB 32-LEN V$;V$
200 NEXT C
210 IF C>N THEN GOTO 440
220 LET V=(V*T(C)/100+V)*1.005
230 GOSUB 330
240 RETURN
250 PRINT "SAQUE(-)/DEPOSITO?";
260 INPUT K$
270 LET S=VAL K$
280 GOSUB 380
290 PRINT TAB 32-LEN S$;S$
300 LET V=V+(S)
310 GOSUB 330
320 RETURN
330 LET V$=STR$ (V+.001)
340 FOR I=1 TO LEN V$
```

```
 350 IF V$(I)<>"." THEN NEXT I
 360 LET V$=V$( TO I+2)+"0"
 370 RETURN
 380 LET S$=STR$ (S+.001)
 390 IF SGN S=-1 THEN LET S$=STR
$ (S-.001)
 400 FOR I=1 TO LEN S$
 410 IF S$(I)<>"." THEN NEXT I
 420 LET S$=S$( TO I+2)+"0"
 430 RETURN
 440 PRINT
 450 PRINT TAB 5;"[illegible]: C
R$";TAB 32-LEN V$;V$
 460 PRINT ,,"NOVOS CALCULOS? (S
/N)"
 470 INPUT R$
 480 IF R$<>"N" THEN GOTO 500
 490 STOP
 500 CLS
 510 GOTO 10
 520 SAVE "POUPANCA"
 530 RUN
```

```
[illegible]? - CR$  14550.000

COR MON 1º MES?: 9.75 %
[illegible]+[illegible]+[illegible]=        16048.460
SAQUE(-)/DEPOSITO?                   .000
COR MON 2º MES?: 8.93 %
[illegible]+[illegible]+[illegible]=        17569.000
SAQUE(-)/DEPOSITO?               -575.250
COR MON 3º MES?: 9.27 %
[illegible]+[illegible]+[illegible]=        18661.920
SAQUE(-)/DEPOSITO?                250.000
COR MON 4º MES?: 8.68 %
[illegible]+[illegible]+[illegible]=        20656.240
SAQUE(-)/DEPOSITO?                   .000
COR MON 5º MES?: 9.49 %
[illegible]+[illegible]+[illegible]=        22729.600
SAQUE(-)/DEPOSITO?                   .000
COR MON 6º MES?: 0 %

     [illegible]: CR$  22729.600

NOVOS CALCULOS? (S/N)
```

Para rodar em FAST, acrescentar as seguintes linhas:

```
35 FAST
455 SLOW
```


O programa foi elaborado para conferências ou cálculos de rendimentos de cadernetas de poupança até 12 meses. Para aumentar esse prazo o valor atribuído à variável N na linha 10 do programa deverá ser modificado.

Caso seja alterada a atual remuneração de juros de 0,5% ao mês, o fator 1.005 da linha 220 deverá ser alterado correspondentemente.

Uma vez introduzido o valor inicial aplicado, o programa irá solicitando através do vídeo as taxas de correção monetária e saques ou depósitos eventuais de cada mês. Introduzidos os dados, via teclado, o saldo acumulado do mês será imediatamente apresentado, corrigido e acrescido dos juros normais. Deve ser lembrado que um depósito feito depois do dia do mês de referência da caderneta só é válido para correção e juros no mês seguinte, devendo portanto ser lançado de acordo.

Para interromper a sequência num determinado mês, bastará digitar 0 (zero) quando o computador solicitar a taxa de correção do mês.

Não havendo saques nem depósito, deverá ser fornecido também o valor 0 quando o computador solicitar esses dados no mês em processamento. Se porventura ocorrerem ambos, deve ser apurada a diferença e introduzida apenas esta, como saque ou depósito, ou seja, a do que prevalecer.

Se o programa puder ser de grande serventia, será aconselhável tê-lo gravado em fita K-7, para utilização rápida a qualquer momento. Antes de fazê-lo, porém, leia o capítulo seguinte a este, MANIPULANDO DIVERSOS PROGRAMAS A UM SO TEMPO, já que poderá ser interessante agrupar na mesma gravação os outros programas financeiros apresentados.

MANIPULANDO DIVERSOS PROGRAMAS A UM SÓ TEMPO

No capítulo GRAVADOR K-7, A OUTRA METADE DO MICROCOMPUTADOR, fizemos alusão ao fato de os microcomputadores chamados de "uso pessoal" não poderem manipular programas complexos e extensos a um só tempo, em razão das limitações de suas memórias, e que produzir os mesmos com memórias agigantadas só se justificaria em casos excepcionais, mesmo porque quase todos podem ter suas memórias "expandidas" através de instalação de memórias adicionais ou implementos apropriados produzidos para esse fim. O TK 85, por exemplo, é vendido atualmente em duas modalidades, de 16 e 48 K "bytes", podendo o primeiro receber expansão posterior mediante solicitação ao fabricante.

Mas, em verdade, nada impede que diversos programas possam ser agrupados e carregados no microcomputador concomitantemente, para manipulação alternada, desde que obedecidas duas regras básicas e normais:

- NÃO EXCEDER A CAPACIDADE DA RAM;

- CADA PROGRAMA TER NUMERAÇÃO DE LINHAS SUPERIOR AO ANTECEDENTE, ISTO É, AO PROGRAMA QUE O PRECEDE.

Alguns poucos cuidados são necessários, por precaução:

- separar um programa do outro com uma linha de instrução de parada, STOP, no fim de cada um;

- se os programas utilizarem variáveis, numéricas ou alfanuméricas, colocar no começo de cada um uma linha de instrução CLEAR, para que não haja problemas no caso de utilização de variáveis análogas nos diversos programas.

- para facilitar a manipulação do conjunto, colocar no início um "menu" e em cada programa uma instrução de retorno opcional ao mesmo.

Com base no exposto, e a título de exemplo, vamos agrupar os programas apresentados no capítulo PROGRAMAS FINANCEIROS, já que os referidos tem finalidades próximas e poderão ser eventualmente usados sempre numa mesma oportunidade. Agrupados e gravados em fita K-7, será fácil e proveitosa sua utilização a qualquer momento.

Façamos primeiramente o menu:

```
   5 CLS
  10 PRINT AT 5,5;"PROGRAMAS FIN
ANCEIROS"
  20 PRINT ,,,,
  30 PRINT TAB 3;"A - APLICACAO
 DE  CAPITAL"
  40 PRINT
  50 PRINT TAB 3;"I - APLICACAO
  PROGRAMADA"
  60 PRINT
  70 PRINT TAB 3;"J - TAXA DE  F
INANCIAMENTO"
  80 PRINT
  90 PRINT TAB 3;"M - VALOR PRES
TACAO MENSAL"
 100 PRINT
 110 PRINT TAB 3;"P - CADERNETA
DE  POUPANCA"
 120 LET E$=INKEY$
 130 IF E$="A" THEN GOTO 200
 140 IF E$="I" THEN GOTO 500
 150 IF E$="J" THEN GOTO 800
 160 IF E$="M" THEN GOTO 1100
 170 IF E$="P" THEN GOTO 1400
 180 IF E$<>"A" OR E$<>"I" OR E$
<>"J" OR E$<>"M" OR E$<>"P" THEN
 GOTO 120
```

Digite agora o programa APLICAÇÃO DE CAPITAL do capítulo mencionado acima, numerando suas linhas a partir de 210, isto é, acrescentando 200 a cada número já existente - para facilidade de controle das alterações - e alterando as instruções e desvios GOTOS e GOSUBS correspondentemente.

Acrescente no mesmo duas linhas iniciais:

```
200 CLS
205 CLEAR
```

Altere a linha de número original 140 para:

340 IF R$="N" THEN GOTO 5

Suprima a linha de número original 150.

Acrescente uma última linha de instrução, por precaução, para e-vitar que os programas se misturem no caso de algum lapso:

380 STOP

Proceda da mesma forma com o programa APLICAÇÃO PROGRAMADA, numerando suas linhas a partir de 510, isto é, acrescentando 500 a cada número original de linha de instrução. Altere também as instruções GOTOS e GOSUBS correspondentemente.

Acrescente ao mesmo duas linhas iniciais:

500 CLS
505 CLEAR

Altere a linha de número original 200 para:

700 IF R$="N" THEN GOTO 5

Elimine a linha de número original 210.

Acrescente a última linha de precaução:

740 STOP

Adote o mesmo procedimento com relação ao programa TAXA DE FINANCIAMENTO, numerando suas linhas a partir de 810, isto é, acrescentando 800 a cada número original de linha do mesmo, não esquecendo de alterar também as instruções de desvios GOTOS e GOSUBS em conformidade.

Introduza no mesmo duas linhas iniciais:

800 CLS
805 CLEAR

Altere a linha de número original 200 para:

1000 IF R$="N" THEN GOTO 5

Elimine a linha de número original 210.

Acrescente a última linha de precaução:

1040 STOP

Faça as alterações necessárias também no programa VALOR DA PRESTAÇAO MENSAL, numerando suas linhas a partir de 1110, isto é, acrescentando 1100 a cada número original de linha de instrução, não esquecendo de fazer o mesmo nas instruções de desvios GOTOS e GOSUBS.

Introduza no mesmo duas linhas iniciais:

1100 CLS
1105 CLEAR

Altere a linha de número original 150 para:

1250 IF R$="N" THEN GOTO 5

Elimine a linha de número original 160.

Acrescente a última linha de precaução:

1290 STOP

Proceda igualmente com o programa CADERNETA DE POUPANÇA, numerando suas linhas a partir de 1410, isto é, acrescentando 1400 a cada número original de linha de instrução, alterando também as linhas de desvios GOTOS e GOSUBS.

Introduza no mesmo duas linhas iniciais:

1400 CLS
1405 CLEAR

Altere a linha de número original 480 para:

```
1880 IF R$="N" THEN GOTO 5
```

Elimine a linha de número original 490.

Confira as alterações feitas, com especial atenção nas linhas de instruções de desvios GOTOS e GOSUBS, e rode o programa para constatar a comodidade de, em certos casos, ter programas agrupados.

Grave em seguida o conjunto assim obtido, fazendo o comando:

GOTO 1920 e NEW LINE.

Caso queira controlar a quantidade de "bytes" ocupados pelos programas agrupados, faça o comando direto:

PRINT PEEK 16404+256*PEEK 16405-16509 e NEW LINE.

Se for agrupar mais algum programa ou acrescentar novas linhas e instruções e estiver se aproximando do limite de memória RAM do computador, o controle pode ser efetuado a cada acréscimo, colocando o comando acima como linha de programa, próximo do número extremo superior do computador, digamos 9990 (o último número possível no TK é 9999), e sempre que quiser aquela informação, basta fazer o comando direto:

GOTO 9990 e NEW LINE.

Agrupamentos de programas que contenham linhas REM com códigos em linguagem de máquina se recomendam apenas a usuários com muita experiência, em razão de que as alterações necessárias, especialmente as dos endereços dos códigos, serem razoavelmente complexas.

Deve ser acrescentado que o agrupamento exemplificado pode ser feito diretamente de programas já gravados em fitas K-7, com auxílio de um programa renumerador e acoplador, este último conhecido como "simulador merge".

SIMULANDO DATA, READ, RESTORE

Tais instruções, existentes em alguns computadores como rotinas residentes, permitem que num programa seja estabelecida uma tabela com dados - DATA - que podem ser obtidos e usados sequencialmente pelo programa, à medida de sua conveniência - através de READ - e, quando a tabela é esgotada, pois permite apenas uma leitura de cada dado, e há necessidade de dispor novamente dela, a mesma é restabelecida através de RESTORE.

No TK essas instruções devem ser programadas.

Para a tabela propriamente dita - DATA - usa-se uma variável alfanumérica qualquer, por exemplo T$, e para que a tabela forneça os dados em sequência e possa, depois de esgotada, ser restabelecida - RESTORE - é usada uma variável comum, por exemplo R, à qual deve ser atribuído um valor inicial 1.

Supondo que os dados da tabela devam ser meses do ano, as instruções terão a seguinte configuração:

```
10 LET T$="JANEIRO/FEVEREIRO/ MARÇO"  etc.
20 LET R=1
```

A leitura da tabela - READ - será feita por uma subrotina do programa, colocada no fim do mesmo, através de uma instrução comum de desvio GOSUB, colocada no ponto onde deve ser feita a leitura de um dado da tabela ou onde for conveniente. Sua configuração será:

```
 100 GOSUB 5000
 ...................   (linhas normais
 ...................         do programa)
5000 FOR Z=R TO LEN T$
5010 IF T$(Z)<>"/" THEN NEXT Z
5020 LET L$=T$(R TO Z-1)
5030 LET R=Z+1
5040 RETURN
```

Os números das linhas de instruções poderão ser quaisquer, desde que seja mantida a sequência indicada, não podendo ser mudada sua composição.

Para melhor compreensão do que foi explanado, é apresentado a seguir um programa com exemplificação:

```
   1 REM  DATA, READ, RESTORE
  10 DIM V(12)
  20 DIM G(12)
  30 LET VT=0
  40 LET GT=0
  50 LET T$="JAN/FEV/MAR/ABR/MAI
/JUN/JUL/AGO/SET/OUT/NOV/DEZ"
  60 LET R=1
  70 FOR C=1 TO 12
  80 PRINT "VENDAS DE ";
  90 GOSUB 1000
 100 PRINT L$;": ";
 110 INPUT V(C)
 120 PRINT V(C)
 130 LET VT=VT+V(C)
 140 CLS
 150 IF R>LEN T$ THEN GOTO 200
 160 NEXT C
 200 LET R=1
 210 FOR C=1 TO 12
 220 PRINT "CUSTO MERCADORIAS ";
 230 GOSUB 1000
 240 PRINT L$;": ";
 250 INPUT G(C)
 260 PRINT G(C)
 270 LET GT=GT+G(C)
 280 CLS
 290 IF R>LEN T$ THEN GOTO 310
 300 NEXT C
 310 PRINT TAB 7;"BÅLANCO PARCIA
L"
 320 PRINT ,,"VENDAS DO ANO";TAB
 32-LEN STR$ VT;VT
 330 PRINT ,,"CUSTO DAS MERCADOR
IAS";TAB 32-LEN STR$ GT;GT
 340 LET T=VT-GT
 350 PRINT ,,"LUCRO BRUTO";TAB 3
2-LEN STR$ T;T
 360 STOP
1000 FOR Z=R TO LEN T$
1010 IF T$(Z)<>"/" THEN NEXT Z
1020 LET L$=T$(R TO Z-1)
1030 LET R=Z+1
1040 RETURN
```


Depois de digitado o programa:

Tecle RUN e NEW LINE. Forneça os valores que o programa solicitar através do vídeo e observe como funcionam as instruções simuladoras de

DATA na linha 50,

READ nas linhas 90, 230, 1000 a 1040 e

RESTORE nas linhas 60 e 200.

Embora à primeira vista um pouco longo para ilustrar o funcionamento das instruções em questão, pretendeu-se também dar através do programa um exemplo prático de sua aplicação, envolvendo outros recursos anteriormente vistos.

Quaisquer dados que tenham que ser repetidos frequentemente e numa mesma sequência num programa podem ser agrupados nos moldes exemplificados. Em certas situações o agrupamento e a utilização de dados, especialmente quando são muitos e variados, devem ser feitos conforme o sistema exposto, a fim de que sejam evitados erros de transcrição. Os mais diversos dados podem ser colocados numa tabela: numéricos ou alfanuméricos, sendo necessário apenas agrupá-los conforme exposto acima.

Uma observação importante: as variáveis utilizadas nas instruções em questão não devem ser usadas em outras instruções do mesmo programa, sob pena de ocorrerem anomalias durante o seu funcionamento.

DELETADOR DE LINHAS RAPIDO

Em computação é frequente a necessidade de deletar linhas de programas, especialmente quando se trata de programas editores ou monitores que, uma vez usados para introduzir códigos em linguagem de máquina, por exemplo, devem ser eliminados na forma de deleção, isto é, sem o uso do comando NEW, para que os códigos introduzidos permaneçam na memória do computador.

Quando um programa editor ou monitor tem reduzido número de linhas, a deleção linha por linha não chega a ser problemática. Mas há programas nessa categoria que chegam a ter tal quantidade de linhas de instruções que sua eliminação sem um deletador efioiente seria com certeza desanimadora, quando não pouco praticável.

Outras vezes, num programa em estudo e montagem queremos eliminar, para substituir, apenas parte do mesmo, a qual, sendo extensa, toma tempo precioso que acaba perturbando o raciocínio e o trabalho, não raramente pondo este a perder em razão de um número de linha digitado imperceptivelmente errado.

Um deletador como o que apresentamos a seguir visa evitar essas dificuldades:

```
 xxx LET DL=PEEK 16425+256*PEEK 16426-51
9500 LET ET=PEEK 16425+256*PEEK 16426
9510 LET A=ET-DL+124
9520 LET B=INT (A/256)
9530 LET A=A-B*256
9540 POKE DL,A
9550 POKE DL+1,B
9560 CLS
9570 PRINT "DELETE A LINHA XXX"
```

No caso de utilização em programas editores ou monitores, o deletador deverá ser incorporado a eles de forma definitiva. Veja o exemplo no capítulo EDITOR ASSEMBLER COM CONFERIDOR. Em outros poderá ser digitado no momento em que é necessitado, já que é constituído por número de linhas relativamente insignificante.

A primeira linha do mesmo, de número xxx qualquer, deverá ser colocada uma linha antes da parte a ser eliminada após o uso, o que geralmente ocorre a partir da primeira linha do programa editor ou monitor. Caso o mesmo já inclua linhas REM para receber códigos em linguagem de máquina ou quaisquer outros dados ou ainda outras linhas iniciais que devam permanecer, a linha xxx deverá ser posicionada em seguida às mesmas, recebendo numeração compatível. Supondo-se que o número da primeira linha a ser eliminada seja 100, a linha xxx poderá receber o número 99, ou mesmo 100, ficando assim:

```
99 LET DL=PEEK 16425+256*PEEK 16426-51
```

A parte restante do DELETADOR deverá ser colocada no fim do programa a ser eliminado, recebendo qualquer numeração superior à do mesmo. Supondo que o número de sua última linha seja 6900, o DELETADOR poderá ser numerado como se segue:

```
7000 LET ET=PEEK 16425+256*PEEK 16426
7010 LET A=ET-DL+124
7020 LET B=INT (A/256)
7030 LET A=A-B*256
7040 POKE DL,A
7050 POKE DL+1,B
7060 CLS
7070 PRINT "DELETE A LINHA XXX"
```

Note que no DELETADOR são usadas duas linhas de instruções muito parecidas com duas variáveis diferentes, DL e ET, não devendo as mesmas ser confundidas.

O programa em processamento - não o DELETADOR - deverá ser usado ou manipulado normalmente. Apenas por precaução, caso sua última linha não seja de retorno ou de parada, deverá ser inserida ali, precedendo a parte final do DELETADOR, uma linha de instrução STOP, por exemplo:

```
6950 STOP
```

para que o prosseguimento do programa não acione casualmente o DELETADOR, eliminando tudo que estiver após a linha 99 ou xxx.

Para usar o DELETADOR, bastará fazer o comando direto:

GOTO 7000

e logo surgirá no vídeo a frase:

DELETE A LINHA XXX

Bastará então digitar 99 e teclar NEW LINE e será feita a eliminação ou deleção imediata de todo o programa que estiver após essa linha, não demorando mais do que dois segundos a execução.

No caso de o DELETADOR ter que ser inserido num programa já carregado ou existente na memória do microcomputador, a linha xxx deverá ser posicionada no ponto escolhido e com numeração compatível, devendo ser seguida de instrução suplementar - GOTO 7000 - na linha imediata. Supondo que a parte a ser eliminada seja da linha 501 até o final, a linha xxx e a complementar deverão ter a seguinte composição:

```
501 LET DL=PEEK 16425+256*PEEK 16426-51
502 GOTO 7000
```

As demais linhas - 7000 a 7070 - serão colocadas no fim do programa e na mesma ordem já descrita, naturalmente lembrando que a instrução da linha 7070, ao mencionar xxx em sua execução, refere-se à linha 501. E para usar o DELETADOR, neste caso, deverá ser feito o comando direto:

GOTO 501

e quando aparecer a frase DELETE A LINHA XXX bastará digitar 501 e teclar NEW LINE para ser efetuada a deleção.

O DELETADOR poderá ser usado também para eliminar um bloco intermediário extenso de linhas, bastando para tanto ser posicionado adequadamente, isto é, a linha xxx antes ou no próprio local da primeira linha a ser eliminada e a última linha da parte final, linha 7070, no local da última linha do bloco a ser eliminado.

CONFERIDOR (MONITOR) DE CODIGOS

Não é raro acontecer: depois de digitado com todo o cuidado e posto a rodar, o programa vai para o ar, isto é, some do computador, isto acontecendo geralmente com programas que contêm blocos em linguagem de máquina.

Por essa razão, é sempre aconselhável gravar em fita K-7 qualquer programa recém-digitado, ANTES DE RODÁ-LO, principalmente se incorporar rotinas ou blocos em linguagem de máquina, além do BASIC. Se acontecer de o programa "sumir" ao ser rodado, poderá ser carregado da fita para o computador para pesquisa e correção.

A parte em BASIC poderá ser conferida diretamente, instrução por instrução, ou linha por linha, se for o caso.

A parte em linguagem de máquina terá que ser conferida com o auxílio de um programa conferidor ou monitor.

No caso de o programa ter sido digitado com o auxílio de um programa editor como o apresentado no capítulo EDITOR ASSEMBLER COM CONFERIDOR, que permite gravá-lo juntamente com o programa, para eliminação posterior, isto é, depois de o programa ter sido testado, a conferência será fácil. Além do mais, para maior comodidade, o EDITOR referido incorpora o DELETADOR DE LINHAS RAPIDO. Em caso contrário, todavia, poderá ser introduzido na parte final do programa, via digitação, o pequeno programa conferidor apresentado a seguir, e feita a verificação:

Digite:

```
9900 LET V=xxx        (xxx = endereço do primeiro "byte"
                         dos códigos a serem conferidos.)
9910 LET P=PEEK V
9920 SCROLL
9930 PRINT V;" : ";CHR$ ( INT (P/16)+28);CHR$ (P-INT
     (P/16)*16+28);" = ";P
9940 PAUSE 4E4
9950 LET V=V+1
9960 GOTO 9910
```

Digite GOTO 9900 e tecle NEW LINE. Aparecerão no vídeo: endereço da posição de memória e o código constante do mesmo, este em notações hexadecimal e decimal.

Aperte qualquer tecla - menos BREAK - para continuar a exibição desses dados em sequência.

Ao encontrar um erro anote num papel o endereço e a correção que deverá ser feita.

Quando terminar a conferência tecle BREAK e faça as correções através do comando POKE. Por exemplo: se o código do endereço 16525 deve ser C6 em vez de 06, verifique na tabela HEXA-DECI qual é o código decimal correspondente - no caso 198 - e digite:

POKE 16525,198 e tecle NEW LINE.

Proceda da mesma forma com os demais códigos errados, se houver. Quando terminar coloque uma linha STOP antes do CONFERIDOR (MONITOR) DE CODIGOS, por exemplo 9890 STOP, e teste o programa corrigido. Se estiver tudo em ordem delete o programa CONFERIDOR e grave definitivamente o programa em processamento.

EDITOR ASSEMBLER COM CONFERIDOR

O programa tem por finalidade introduzir no computador códigos hexadecimais ou decimais para produzir rotinas ou programas em linguagem de máquina - linhas ou blocos ASSEMBLER - e conferir ou verificar os códigos introduzidos ou já existentes.

O programa incorpora o DELETADOR DE LINHAS RAPIDO, apresentado e comentado em capítulo de mesmo nome, de forma a poder ser apagado facilmente no computador após o uso, sem afetar os códigos já introduzidos. Incorpora também um produtor de LINHAS REM para quaisquer quantidades de caracteres, de grande serventia para editar programas em linguagem de máquina.

A numeração de linhas do EDITOR ASSEMBLER COM CONFERIDOR a partir de 9000 permite que, após a introdução dos códigos de linhas ou de blocos em ASSEMBLER de um programa, seja digitada também a listagem BASIC que normalmente acompanha e complementa tal tipo de programa, para que o mesmo seja testado antes da eliminação do programa editor, de maneira que possa ser feita com facilidade uma conferência dos códigos introduzidos, se houver falha que não resida na listagem em BASIC.

E oportuno lembrar aqui a conveniência de gravar em fita K-7 um programa recém-digitado, principalmente se contiver instruções em linguagem de máquina, já que qualquer falha nas mesmas poderá fazer ir facilmente "para o ar" todo o programa, caso em que a gravação feita permitirá recolocá-lo no computador para pesquisa e correção da falha.

Depois de testado e aprovado o programa em montagem, o programa EDITOR ASSEMBLER COM CONFERIDOR poderá ser todo eliminado com a fácil deleção de uma única linha, como será constatado.

A primeira parte do mesmo consiste de um "menu", onde a modalidade de execução é escolhida. Seus sete itens são:

1 - INTRODUZ CODIGOS HEXADECIMAIS

2 - INTRODUZ CODIGOS DECIMAIS
3 - CONFERE CODIGOS HEXADECIMAIS
4 - CONFERE CODIGOS DECIMAIS
5 - GRAVA EDITOR/CONFERIDOR
6 - PRODUZ LINHAS REMS
7 - ELIMINA EDITOR/CONFERIDOR

A opção número 5 efetua não somente a gravação do programa EDITOR, mas também a do que estiver em montagem, caso seja necessária uma interrupção para continuação posterior.

Segue-se a listagem do programa, feita por impressora:

```
   1 REM ..
9000 STOP
9005 LET DL=PEEK 16425+256*PEEK
16426-57
9010 CLS
9015 PRINT AT 5,1;"EDITOR ASSEMB
LER COM CONFERIDOR"
9020 PRINT ,,,,
9025 PRINT TAB 5;"1 - INTRODUZ C
OD. HEX."
9030 PRINT
9035 PRINT TAB 5;"2 - INTRODUZ C
OD. DEC."
9040 PRINT
9045 PRINT TAB 5;"3 - CONFERE  C
OD. HEX."
9050 PRINT
9055 PRINT TAB 5;"4 - CONFERE  C
OD. DEC."
9060 PRINT
9065 PRINT TAB 5;"5 - GRAVA EDIT
OR/CONF."
9070 PRINT
9075 PRINT TAB 5;"6 - PRODUZ LIN
HAS REMS"
9080 PRINT
9085 PRINT TAB 5;"7 - ELIMINA ED
IT/CONF."
9090 LET E$=INKEY$
9095 IF CODE E$<29 OR CODE E$>35
 THEN GOTO 9090
9099 GOTO VAL E$*100+9000
9100 CLS
9105 SCROLL
9110 PRINT "ENDERECO?"
9115 INPUT E
9120 LET C$=""
9125 SCROLL
9130 PRINT E;"-/-";
9135 IF PEEK 16441=1 THEN GOTO 9
125
9140 IF C$="" THEN INPUT C$
9145 IF C$="E" THEN GOTO 9105
9150 IF C$="M" THEN GOTO 9005
9155 PRINT C$( TO 2);"-";
9160 POKE E,16*CODE C$+CODE C$(2
)-476
9165 LET E=E+1
9170 LET C$=C$(3 TO )
9175 GOTO 9135
9200 CLS
9205 PRINT TAB 0;"ENDERECO?"
9210 INPUT E
9215 PRINT TAB 0;E;" - ";
9220 FOR C=E TO E+5
9225 INPUT C$
9230 IF C$="E" THEN GOTO 9205
9235 IF C$="M" THEN GOTO 9005
9240 PRINT C$;"-";
9245 POKE C,VAL C$
9250 NEXT C
9255 LET E=C
9260 GOTO 9215
9300 CLS
9305 PRINT "ENDERECO INICIAL? ";
9310 INPUT E
9315 PRINT E
9320 PRINT "ENDERECO FINAL? ";
9325 INPUT F
9330 PRINT F
9335 PRINT
9340 PRINT TAB 0;E;"-/-";
9345 FOR C=E TO E+7
9350 IF C>F THEN GOTO 9380
9355 LET P=PEEK C
9360 PRINT CHR$ (INT (P/16)+28);
CHR$ (P-INT (P/16)*16+28);"-";
9365 NEXT C
9370 LET E=C
9375 GOTO 9340
9380 PRINT TAB 0
9385 PRINT "MENU OU CONFERIR MAI
S? (M/C)"
9390 LET E$=INKEY$
9391 IF E$="M" THEN GOTO 9005
9392 IF E$="C" THEN GOTO 9300
9393 IF E$<>"M" OR E$<>"C" THEN
GOTO 9390
9400 CLS
9405 PRINT "ENDERECO INICIAL? ";
```

```
9410 INPUT E
9415 PRINT E
9420 PRINT "ENDERECO FINAL? ";
9425 INPUT F
9430 PRINT F
9435 PRINT
9440 PRINT TAB 0;E;" - ";
9445 FOR C=E TO E+5
9450 IF C>F THEN GOTO 9475
9455 PRINT PEEK C;"-";
9460 NEXT C
9465 LET E=C
9470 GOTO 9440
9475 PRINT TAB 0
9480 PRINT "MENU OU CONFERIR MAI
S? (M/C)"
9485 LET E$=INKEY$
9486 IF E$="M" THEN GOTO 9005
9487 IF E$="C" THEN GOTO 9400
9488 IF E$<>"M" OR E$<>"C" THEN
GOTO 9485
9500 SAVE "EDITOR"
9510 GOTO 9005
9600 CLS
9610 PRINT "QUANTOS CARACTERES?"
9620 INPUT R
9630 LET S=INT (R/256)
9640 LET R=R-S*256
9650 POKE 16514,R
9660 POKE 16515,S
9670 LET V=USR 8192
9680 CLS
9690 LIST 1
9700 LET ET=PEEK 16425+256*PEEK
16426
9710 LET A=ET-DL+127
9720 LET B=INT (A/256)
9730 LET A=A-B*256
9740 POKE DL,A
9750 POKE DL+1,B
9760 CLS
9770 PRINT ,,"DELETE A LINHA 900
0"
```

Depois de ter digitado e conferido todo o programa:

Digite: GOTO 9005 e tecle NEW LINE. Aparecerá no vídeo o "menu" do programa.

Apronte e acione o gravador para gravar, de preferência com uma fita K-7 virgem e de boa qualidade.

Pressione uma vez a tecla 5 do microcomputador e aguarde o término da gravação do EDITOR ASSEMBLER COM CONFERIDOR.

Faça agora algumas experiências com o programa, lembrando que códigos para rotinas ou programas em linguagem de máquina devem ser colocados em linhas REM ou em área reservada através da manipulação da variável RTP do sistema. Não tente introduzir códigos se não tiver espaço reservado para os mesmos. Vamos fazê-lo nesta oportunidade através do próprio programa EDITOR:

Pressione a tecla 6 do microcomputador. Surgirá no vídeo a pergunta "QUANTOS CARACTERES?".

Digite 100 e tecle NEW LINE. Em seguida:

Digite 1 e tecle NEW LINE. Permanecerá como primeira linha do programa a linha 2 REM com 100 caracteres "." (ponto).

Procedimentos de experiência:

- digite GOTO 9005 e tecle NEW LINE. Aparecerá o "menu".
- pressione a tecla 3. Surgirá uma pergunta no vídeo.
- digite 16514 e tecle NEW LINE. Surgirá outra pergunta.
- digite 16550 e tecle NEW LINE. O vídeo exibirá 37 códigos hexadecimais 1B e seus respectivos endereços na memória do microcomputador. 1B é o código hexadecimal do carácter "." da linha REM. Após os mesmos haverá uma pergunta no vídeo.
- pressione a tecla M. Se resolver pressionar C, repita o processo e depois tecle M. Surgirá o "menu" no vídeo. Mude de ítem: pressione a tecla 4 e note outra pergunta no vídeo.
- digite 16514 e tecle NEW LINE. Nova pergunta no vídeo.
- digite 16560 e tecle NEW LINE. O vídeo exibirá 47 códigos decimais 27, que é o código do carácter N da linha REM. Após estes haverá uma pergunta.
- pressione a tecla M. De novo o "menu" estará no vídeo.
- pressione a tecla 1. Surgirá uma pergunta.

- digite 16514 e tecle NEW LINE. Surgirá o CURSOR L no canto inferior esquerdo do vídeo e o número 16514.
- digite cinco ou dez códigos hexadecimais B3, sem espaço entre os mesmos, e tecle NEW LINE. Os códigos surgirão ordenados após o endereço mencionado e separados por hífens.
- tecle M e NEW LINE. O "menu" estará no vídeo.
- tecle 2. Surgirá de novo uma pergunta.
- digite 16525 e tecle NEW LINE. Surgirá o CURSOR L no canto inferior esquerdo do vídeo e o número 16525 no canto superior.
- digite dez códigos decimais 148, um por vez, teclando NEW LINE após cada um. Os códigos serão exibidos no vídeo após o endereço indicado.
- tecle M e NEW LINE. O "menu" estará no vídeo.
- tecle BREAK para interromper o programa.
- digite LIST e tecle NEW LINE. Observe como ficou a linha 2 REM com os códigos B3 (hexadecimal) e 148 (decimal) introduzidos.
- digite GOTO 9005 e tecle NEW LINE.

Faça mais experiências com o programa EDITOR ASSEMBLER COM CONFERIDOR. Familiarize-se com todos os recursos do mesmo, respondendo alternativamente às perguntas exibidas no vídeo.

Nas opções 1 e 2, quando estiver introduzindo códigos e desejar voltar a um endereço para alteração ou introdução de código diferente do que consta ali, digite E em lugar do código e indique a seguir o endereço a ser acessado. Digite o novo código, tecle NEW LINE e digite E novamente para voltar ao endereço em que estava, indicando seu número via digitação.

MONTANDO DIVERSAS LINHAS REM: Supondo que se queira montar diversas linhas REM, por exemplo: a primeira com 100 caracteres, a segunda com 150 e a terceira com 200, proceda da seguinte forma, após o programa ter sido carregado no microcomputador:

1 - Após o "menu" do programa estar no vídeo, pressione a tecla 6 do microcomputador. Ao surgir a pergunta "QUANTOS CARACTERES?":

2 - Digite 200 e tecle NEW LINE. Surgirá no vídeo uma linha REM com 200 caracteres ".", após a linha 1 REM.

3 - Digite 1 REM .. e tecle NEW LINE. (Deverá ser exatamente como citado: 1 REM .. (com dois caracteres ".")).

4 - Pressione as teclas SHIFT e 6 simultaneamente.

5 - Pressione as teclas SHIFT e EDIT simultaneamente.

6 - Pressione as teclas SHIFT e RUBOUT simultaneamente.

7 - Digite 4 e tecle NEW LINE.

8 - Digite 2 e tecle NEW LINE.

9 - Digite GOTO 9005 e tecle NEW LINE.

10 - Repita os passos 1 a 8, digitando 150 no passo 2 e digitando 3 no passo 7.

11 - Digite GOTO 9005 e tecle NEW LINE.

12 - Digite 100 à pergunta "QUANTOS CARACTERES?" e tecle NEW LINE.

13 - Digite 1 e tecle NEW LINE.

14 - Digite GOTO 9005 e tecle NEW LINE.

15 - Passe a usar o programa EDITOR de acordo com os recursos oferecidos pelo "menu".

O programa em questão poderá ser muito útil desde introduzir códigos para rotinas ou programas em linguagem de máquina até verificar os códigos existentes em todos os endereços do computador, tanto na memória RAM como na ROM.

Faça duas ou três gravações do mesmo. Sua utilização poderá ser bastante frequente.

JOYSTICK DESENHADOR

Títulos, letreiros, molduras, paisagens e o que for imaginado poderão ser desenhados apenas com o uso do "joystick", ou das teclas correspondentes, com facilidade e rapidez, e armazenados em variáveis alfanuméricas, para as mais diversas utilizações. E o que o programa que apresentamos neste capítulo permite fazer.

A reprodução apresentada a seguir, copiada diretamente do vídeo pela impressora, é um exemplo do que pode ser feito pelo mesmo:

O programa consta de uma pequena subrotina em linguagem de máquina e de uma parte em linguagem BASIC.

Os códigos em linguagem de máquina entram em primeiro lugar. Para tanto, deverá ser montado um pequeno programa editor:

Digite:

```
 3 REM ........................ (24 pontos)
10 FOR C=16514 TO 16537
20 INPUT D
30 SCROLL
40 PRINT D
50 POKE C,D
60 NEXT C
```

Tecle RUN e NEW LINE. Estando o CURSOR L no canto inferior do vídeo, lado esquerdo, introduza os códigos decimais que se seguem, um a um, com atenção para evitar qualquer erro de digitação. Tecle NEW LINE após digitar cada código. As barras não devem ser digitadas, servindo apenas para separá-los:

42/16/64/1/6/0/9/235/42/12/64/6/22/35/197/1/32/0/237/176/193/16/246/201

Tecle NEW LINE ao terminar.

Delete em seguida as linhas 10 a 60.

Sem alterar ou mexer na linha 3, introduza a listagem BASIC:

```
   3 REM E(RND  FOR E£RND -7V
AL  4  GOSUB  AT ( PLOT TAN
   5 DIM T$(704)
  10 LET T=250
  20 LET A=30
  30 LET B=20
  40 PLOT A,B
  50 LET A=A-(INKEY$="5" AND A>0
)+(INKEY$="8" AND A<63)
  60 LET B=B-(INKEY$="6" AND B>0
)+(INKEY$="7" AND B<43)
  70 IF INKEY$="0" THEN GOTO 100
  80 IF INKEY$="1" THEN GOTO 200
  90 GOTO 40
 100 PLOT A,B
 110 UNPLOT A,B
 120 LET A=A-(INKEY$="5" AND A>0
)+(INKEY$="8" AND A<63)
```

```
130 LET B=B-(INKEY$="6" AND B>0
)+(INKEY$="7" AND B<43)
140 IF INKEY$="0" THEN GOTO 40
150 IF INKEY$="1" THEN GOTO 200
160 GOTO 100
200 RAND USR 16514
210 GOSUB T
220 LET T=T+20
230 CLS
240 GOTO 20
250 LET A$=T$
260 RETURN
270 LET B$=T$
280 RETURN
290 LET C$=T$
300 CLS
350 PRINT AT 5,5;"APERTE QUALQU
ER TECLA            PARA VER  OS
DESENHOS"
360 PAUSE 4E4
370 CLS
380 PRINT A$
390 PAUSE 4E4
400 CLS
410 PRINT B$
420 PAUSE 4E4
430 CLS
440 PRINT C$
450 PAUSE 4E4
460 CLS
470 PRINT ,,"VER DE NOVO (V)/FA
ZER NOVOS (F)?"
480 LET R$=INKEY$
490 IF R$="" THEN GOTO 480
500 IF R$="V" THEN GOTO 350
510 IF R$="F" THEN CLEAR
520 CLS
530 GOTO 5
540 SAVE "DESENHOS"
550 RUN
```

Confira e faça uma gravação do programa antes de rodá-lo. Se houver um erro na subrotina em linguagem de máquina o programa poderá "ir para o ar", caso em que deverá ser carregado da fita para ser procurada e corrigida, através do comando POKE, a falha porventura existente. Se for necessário use o programa CONFERIDOR (MONITOR) DE CODIGOS, apresentado no capítulo de mesmo nome. Se estiver tudo em ordem:

Tecle RUN e NEW LINE para começar a "pintar" no vídeo. O pequeno "pixel" preto (carácter de PLOT) que aparece no centro da tela de TV

poderá ser movido para quaisquer direções com o uso do "joystick" - ou das teclas correspondentes - desenhando no vídeo o que for desejado.

Para APAGAR, CORRIGIR OU ALTERAR o desenho:

- APERTE O DISPARADOR DO "JOYSTICK" OU A TECLA 0 e o "pixel" tornar-se-á piscante, transformando-se em APAGADOR. Movimente-o para apagar da mesma forma que para desenhar.

Para RETORNAR A MODALIDADE DE DESENHADOR:

- APERTE DE NOVO O DISPARADOR DO "JOYSTICK" OU A TECLA 0.

Quando der o desenho por acabado, pressione uma vez a tecla 1, o que fará com que o desenho seja ARQUIVADO NA VARIAVEL A$ reservada na linha 250 do programa.

O PROCESSO SERA ENTÃO REINICIADO PARA NOVO DESENHO, COM O PEQUENO "PIXEL" NO CENTRO DO VIDEO ESPERANDO SER DIRIGIDO PARA DESENHAR.

Caso não deseje fazer outro desenho, pressione apenas a tecla 1. A segunda tela será então arquivada em branco e, novamente, o processo se reiniciará.

Ao final da terceira tela, após ter sido pressionada a tecla 1, o programa passará a exibir os desenhos feitos ou as telas em branco arquivados, bastando para tanto apertar qualquer tecla do computador, menos BREAK, perguntando em seguida se se deseja ver os mesmos novamente ou se se quer fazer novos, e procederá em conformidade com nossa ordem, dada via teclado: exibirá as telas já feitas ou as destruirá, colocando novamente no vídeo o "pixel" para produzir novos desenhos.

O programa tem variáveis dimensionadas para três telas somente. Se houver necessidade de mais, bastará acrescentar duas linhas de instruções para cada, nos moldes das linhas 250 e 260, incrementando sequencialmente os números de linhas de instruções e as letras das variáveis alfanuméricas, procedendo correspondentemente com respeito às linhas de instruções para exibição das telas.

No caso de desejar usar os desenhos feitos em outros programas, delete o presente, isto é, o programa desenhador, com auxílio do dele-

tador rápido apresentado no capítulo DELETADOR DE LINHAS RAPIDO, tendo cuidado em não usar o comando ou instrução RUN para não "apagar" os desenhos arquivados nas variáveis.

Digite em seguida o programa em que queira usar os desenhos, inserindo nos pontos em que os mesmos devem ser introduzidos as instruções que os buscam no arquivo das variáveis, ou sejam: PRINT A$, PRINT B$, PRINT C$ etc.

Apresentamos a seguir um pequeno programa exemplo para uma aplicação curiosa do que foi explanado.

Depois de ter "pintado" ou desenhado três lindas telas e de ter deletado o programa desenhador, distraia-se um pouco admirando suas obras de arte:

Digite:

```
 10 PRINT A$
 20 FOR C=1 TO 50
 30 NEXT C
 40 CLS
 50 PRINT B$
 60 FOR C=1 TO 50
 70 NEXT C
 80 CLS
 90 PRINT C$
100 FOR C=1 TO 50
110 NEXT C
120 CLS
130 GOTO 10
```

Tecle GOTO 10 e NEW LINE. NÃO USE RUN PARA NÃO DESTRUIR AS TELAS ARQUIVADAS. Como sabemos, o comando RUN cancela todas as variáveis utilizadas e as telas ficam arquivadas em variáveis alfanuméricas.

Se desejar conservar as telas, grave-as em fitas, acrescentando ao pequeno programa acima:

```
140 SAVE "TELAS"
150 GOTO 10
```

Quando o gravador estiver acionado para gravar, faça o comando direto:

GOTO 140 e NEW LINE.

Use o programa JOYSTICK DESENHADOR para fazer títulos e desenhos para aplicações em jogos. Estes poderão se tornar muito mais interessantes e movimentados com auxílio do programa desenhador.

O programa desenhador poderá ser modificado para produzir também - naturalmente através do teclado - textos para processamento, conjunto ou não, pelo mesmo sistema.

Apresentamos a seguir mais um desenho, feito com auxílio do programa desenhador e baseado numa propaganda do próprio TK, tendo sido copiado diretamente do vídeo pela impressora:

DUPLICANDO PROGRAMAS FECHADOS

Uma grande parte dos programas comerciais, se não a maioria, sejam jogos ou utilitários, é "protegida" contra duplicações, com o intuito de evitar "pirataria".

São programas dotados de artifícios camuflados, os mais diversificados, que os "fecham" ou "travam" de tal maneira que se torna impossível, em alguns casos, até examinar sua listagem para fins de estudos ou curiosidade. Alguns, ao serem carregados da fita K-7 para o computador, já saem "rodando", isto é, entram imediatamente em execução, sendo impossível pará-los com qualquer dos comandos disponíveis. São eliminados da memória do computador apenas se este é desligado. Outros, quando aceitam um BREAK ou um STOP, emperram, "vão para o ar", ou simplesmente deixam de funcionar se forem novamente acionados pelo comando RUN. Alguns voltam a rodar pelo comando GOTO, dependendo de se descobrir para onde mandar o comando, mas não são listados integralmente por força de outros artifícios.

Enfim, uma série infindável de recursos é adotada por programadores ou produtores com a finalidade de dificultar duplicações, cópias e imitações fáceis.

Embora justificáveis, tais medidas acabam por prejudicar o usuário sério quando, por acidente ou em consequência de qualquer problema no gravador, ou ainda por ocorrência ulterior de defeito na fita, qualquer programa de custo elevado se torna imprestável, obrigando a uma nova aquisição. Conscientes de tais possibilidades e visando minimizá-las, alguns poucos produtores fornecem suas fitas com os programas duplamente gravados, uma gravação em cada lado da fita.

Mas, se é infindável a procura e adoção de novos recursos para proteção de programas, incansável é a atividade de estudiosos e curiosos em sentido contrário, isto é, visando "abrir" ou simplesmente duplicar com perfeição tais programas. Assim é que muitos recursos têm sido descobertos e se tornam válidos, sendo praticamente impossível relacionar e descrever todos, já que cada um é um caso particular e depende das peculiaridades de cada programa.

Existem alguns, todavia, de caráter "universal", que funcionam na quase totalidade dos programas e que são aqui publicados com finalidade didática e ilustrativa, ficando sua utilização sujeita à responsabilidade de cada um.

Um deles, muito simples e eficiente, consiste em utilizar direta, mas apenas parcialmente, a própria subrotina LOAD da memória ROM. Seu endereço inicial é 832. O comando feito a partir do endereço 837 permite "abrir" praticamente qualquer programa. Os procedimentos a adotar são os seguintes:

1 - Por o microcomputador na modalidade FAST:

Digite FAST e tecle NEW LINE.

2 - Acionar a subrotina LOAD no ponto mencionado:

Digite RAND USR 837 e tecle NEW LINE.

3 - Acionar o gravador em PLAY com o programa a ser carregado e acompanhar na TV os sinais característicos de carregamento.

Quando for completado o carregamento no computador, aparecerá no canto inferior esquerdo do vídeo a notação C/0, que, não obstante ser C indicação de irregularidade - o que ocorre em decorrência da subrotina ter sido acionada em ponto avançado em relação a seu endereço inicial - significa que o programa foi carregado.

4 - NÃO TECLAR RUN OU LIST para rodar ou listar o programa. Teclar simplesmente NEW LINE, o que deverá fazer aparecer parte da listagem, no mínimo.

Se, por feliz casualidade, encontrar-se nessa parte a linha que contenha a instrução SAVE, execute o passo a seguir, de número 5. Caso contrário, terá que ser tentada sua localização com auxílio do comando LIST, percorrendo sequencialmente o programa até a mesma ser localizada. Se o programa "não for para o ar" ou não "emperrar".

5 - Acionar o gravador para gravar - REC e PLAY -, de preferência com fita virgem.

6 - Acionar em seguida o microcomputador.

Digite GOTO xxx (xxx = nº da linha da instrução SAVE) e tecle NEW LINE.

Logo surgirão no vídeo os sinais característicos do processo de gravação. Após a mesma completar-se, o programa deverá estar "rodando" no computador. Deverá então ser acionada a tecla STOP do gravador e adotado o procedimento normal com o computador.

Outro recurso consiste nos seguintes procedimentos:

1 - Baixar a RTP para o endereço 32753:

Digite POKE 16388,241 e tecle NEW LINE.

Digite POKE 16389,127 e tecle NEW LINE.

Tecle NEW e em seguida NEW LINE.

2 - Introduzir instruções acima da RTP que permitam a duplicação do programa. As instruções são introduzidas através do programa que se segue:

Digite:
```
10 LET A$="CD4403CDBB022C28FACD230FC3F502"
20 LET A=32753
30 FOR C=1 TO LEN A$-1 STEP 2
40 POKE A,16*(CODE A$(C)-28)+CODE A$(C+1)-28
50 LET A=A+1
60 NEXT C
```

Confira bem os códigos da linha 10 e o programa.

Tecle RUN e NEW LINE. Depois de aparecer a notação 0/0 no canto inferior esquerdo do vídeo:

Tecle NEW e NEW LINE. Após aparecer o CURSOR K no canto inferior esquerdo do vídeo:

Tecle FAST e NEW LINE e, em seguida:

Digite RAND USR 32753 e tecle NEW LINE.

3 - Acionar o gravador em PLAY com o programa a ser duplicado. Logo surgirão no vídeo os sinais característicos do processo LOAD.

Quando for completado o carregamento do programa no computador o vídeo ficará mais escuro - acinzentado - sem qualquer notação.

4 - Acionar o gravador para gravar - REC e PLAY -, de preferência com fita virgem e de boa qualidade.

5 - Pressionar em seguida qualquer tecla do microcomputador - menos BREAK - e atentar para os sinais característicos de gravação que surgirão no vídeo. Aguardar o término do processo. Ao cessarem as listras no vídeo deverá surgir no mesmo, no canto inferior esquerdo, a notação C/0. Acione a tecla STOP do gravador e proceda normalmente com o programa no computador.

Observação: Um programa duplicado segundo este processo, ao ser feito seu carregamento para o computador, entrará apenas listado e na modalidade FAST, não obstante poder conter a instrução RUN após SAVE, apresentando-se o vídeo acinzentado e sem qualquer notação. Bastará então teclar BREAK e em seguida SLOW. Uma gravação normal poderá ser feita, então, a partir da instrução SAVE do próprio programa, para que, a partir dessa gravação, o mesmo entre rodando normalmente em SLOW após o seu carregamento.

DUPLICANDO PROGRAMAS ATRAVES DE DLOAD E DSAVE

Como dissemos no início do capítulo anterior, em matéria de programas "fechados" ou "protegidos" cada caso é um caso. Não obstante os recursos ali expostos para se obter duplicações de tais programas serem considerados "infalíveis" por "experts", podem os mesmos esbarrar com alguns novos "truques" que surjam, mais difíceis de serem desvendados. Por isso, apresentamos mais um programa para a mesma finalidade, com recursos mais amplos e que introduzem duas novas funções do TK 85. Com a utilização de tal programa é praticamente impossível não conseguir a duplicação de qualquer programa, por mais protegido que seja.

O programa utiliza as funções DLOAD e DSAVE, só recentemente implantadas no TK, a partir do modelo 85, com finalidades bastante abrangentes, especialmente em matéria de processamento de arquivos.

Através de DLOAD podemos colocar numa área reservada, praticamente do tamanho da memória RAM, qualquer programa que esteja dentro desse limite, incluído nele as poucas instruções necessárias para a finalidade e os "bytes" ocupados pelo arquivo de imagem do computador. E através de DSAVE podemos recuperar o programa ali colocado, duplicado para funcionamento normal.

As instruções necessárias para essa finalidade ocupam, entre outras, duas variáveis chamadas "reservadas", porque especialmente definidas para esse propósito, Z$ e Z, as quais, todavia, quando não utilizadas com as funções DLOAD e DSAVE, são usadas normalmente como as demais do computador.

Estudar essas funções agora seria prematuro. Vamos apenas fazer um pequeno uso das mesmas, de forma que fique demonstrada sua enorme potencialidade. DLOAD se inicia no endereço 8305 e DSAVE no 8288, sendo acessadas através do comando USR.

Segue-se o pequeno programa com as instruções necessárias:

Digite:

```
10 DIM C$(15000)     (Em vez de 15000 podem ser apenas
20 LET Z=0            os "bytes" suficientes para rece
30 LET Z$="C"         ber o programa a ser duplicado.)
40 LET C=USR 8305
50 PRINT "CARREGAMENTO ";"FEITO" AND NOT C;
   "FALHOU" AND C
60 PAUSE 4E4
70 LET C=USR 8288
80 PRINT "COPIA ";"FEITA" AND NOT C;"FALHOU" AND C
```

Obs.: AND (tecla 2) e NOT (tecla N) são comandos de ação indireta. Devem ser usados através das teclas próprias e não digitados letra por letra.

Procedimentos:

- coloque no gravador a fita com o programa a ser duplicado;
- faça as conexões EAR do micro e do gravador;
- tecle RUN e NEW LINE;
- acione a tecla PLAY do gravador;
- espere aparecer no vídeo "CARREGAMENTO FEITO". Se porventura aparecer "CARREGAMENTO FALHOU", o processo deve ser reiniciado;
- acione a tecla STOP do gravador;
- tire a fita lida do gravador e coloque no mesmo uma fita, nova de preferência;
- faça as conexões das tomadas MIC do micro e do computador;
- acione as teclas REC e PLAY do gravador;
- pressione qualquer tecla do computador, menos BREAK;
- espere aparecer no vídeo "COPIA FEITA";
- se desejar mais de uma cópia do mesmo programa, deixe o gravador rodando em REC e PLAY, digite GOTO 70 e tecle NEW LINE no computador.

Se o carregamento - DLOAD - foi efetuado com sucesso, é muitíssimo pouco provável que ocorra qualquer falha em DSAVE, de maneira que a duplicação deve ter sido completada com absoluto sucesso. Todavia, se

ocorrer alguma falha na primeira tentativa, confira todos os ítens do programa e verifique se estão certas as posições dos controles de volume e tom do gravador, quando for efetuar o carregamento do programa a ser duplicado, na próxima tentativa, e certifique-se de que a conexão entre o gravador e o microcomputador esteja bem feita através do cabo coaxial apropriado.

Se o programa a ser duplicado não for maior do que a área reservada na memória do microcomputador, não poderá haver falha.

CONTROLANDO AS QUANTIDADES DE "BYTES" USADOS

Saber as quantidades de "bytes" usados ou disponíveis, por diversas e até muitas razões, é uma imposição constante com que o programador ou usuário de computador logo se defronta, decorrendo daí a necessidade de ter sempre em mente, ou facilmente acessíveis, dados que lhe permitam fazê-lo.

Para saber a quantidade de "bytes":

- DE UMA INSTRUÇÃO OU PROGRAMA NA RAM:

Digite: PRINT PEEK 16396+256*PEEK 16397-16509

Tecle NEW LINE.

- DE UM PROGRAMA MAIS O ARQUIVO DE IMAGEM:

Digite: PRINT PEEK 16400+256*PEEK 16401-16509

Tecle NEW LINE.

- DE UM PROGRAMA, ARQUIVO DE IMAGEM E VARIAVEIS UTILIZADAS:

Digite: PRINT PEEK 16404+256*PEEK 16405-16509

Tecle NEW LINE.

- DISPONIVEIS ALEM DA AREA OCUPADA (RAM DE 16 K "BYTES"):

Digite: PRINT 32768-(PEEK 16404+256*PEEK 16405)

Tecle NEW LINE.

Se for muito frequente a necessidade de ter qualquer dessas informações durante a digitação de um programa longo, por exemplo, coloque a instrução adequada numa linha de programa com numeração próxima do limite superior (o número máximo de linhas no TK 85 é 9999 como sabemos):

```
9995 PRINT PEEK 16404+256*PEEK 16405-16509
```

e toda vez que quiser a informação correspondente, faça o comando mais simples e direto:

GOTO 9995 e tecle NEW LINE.

Se desejar, além desse dado, saber com a mesma frequência quanto resta de memória utilizável, acrescente:

```
9996 PRINT 32768-(PEEK 16404+256*PEEK 16405-16509)
```

Para melhor compreensão dos comandos acima, leia o capítulo ORGANIZAÇÃO DE MEMÓRIA DO TK 85 E AS VARIÁVEIS DO SISTEMA.

UM TESTE EFETIVO DA MEMÓRIA DO COMPUTADOR

Às vezes acontece: quando se vai adiantado na digitação trabalhosa de um programa - e quando acontece geralmente é com um programa longo - de repente as coisas começam a ficar confusas no vídeo, linhas do programa saltam umas às outras, números e caracteres aparecem e desaparecem em turbilhão, a tela de TV fica acinzentada, a desordem impera e o programa "vai para o ar", "some" das vistas nossas e do vídeo.

E aí, tudo tem que ser começado de novo e, mais uma vez, o problema ocorre, causando-nos tremenda frustração e aborrecimentos.

Surgem as dúvidas. Uma tecla pressionada indevidamente? O comando RUN acionado fora de tempo? Enfim, uma série de indagações é feita e não se chega a nenhuma conclusão, simplesmente porque, muito provavelmente, poderá ter ocorrido um desarranjo na memória do computador. Por essa razão, será sempre conveniente efetuar-se esporadicamente um teste de avaliação das condições da memória do aparelho. O pequeno programa apresentado a seguir tem essa finalidade.

Digite:

```
10 FAST
20 LET A=193
30 FOR E=17550 TO 32700
40 POKE E,A
50 NEXT E
60 FOR C=17550 TO 32700
70 IF PEEK C<>A THEN PRINT C,PEEK C
80 NEXT C
90 PRINT "TESTE FEITO"
```

Tecle RUN e NEW LINE. Aguarde aproximadamente quatro minutos, durante os quais a tela de TV ficará apenas acinzentada.

Em silêncio, o microcomputador estará fazendo um "checkup" minucioso em sua própria memória, endereço por endereço. E logo informará, através da TV, seu diagnóstico.

Decorrido o tempo mencionado, se aparecer no vídeo somente a frase TESTE FEITO, estará tudo em ordem com a memória do computador.

Todavia, se aparecerem números entre 17550 e 32700, seguidos de códigos diferentes de 193, é indicação de que os endereços ou "bytes" correspondentes não estão funcionando perfeitamente, carecendo de aferimento pelo departamento de assistência técnica do fabricante ou de serviço técnico autorizado.

A variável A da linha 20 do programa não precisa necessariamente ser definida com o valor 193. Poderá ser qualquer valor entre 0 e 255.

PROGRAMAS

Neste capítulo apresentaremos alguns programas, em listagens feitas por impressora, os quais, além da finalidade prática - exceção feita ao que se destina a entretenimento - oferecem ao leitor usuário de microcomputador interessado em programação exemplos de extensa e variada gama de recursos aplicáveis em computação, através dos quais são obtidos os mais diversos resultados e efeitos.

Os seguintes programas serão apresentados:

- GRAFICO COMPARATIVO DE VENDAS.
- CADASTRO DE LIVROS.
- CONTROLE DE ESTOQUE.
- PROGRAMAS FINANCEIROS.
- BATALHA AR E MAR.

As listagens dos mesmos serão precedidas de um ou mais quadros de vídeo produzidos pelo programa respectivo, seja um gráfico estatístico, um "menu" ou uma apresentação, de forma a tornar tão inteligível quanto possível cada um.

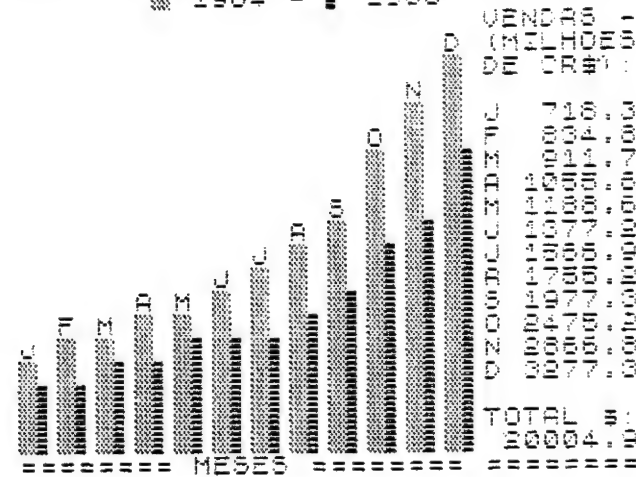
Depois de digitados e gravados em fita K-7 através das respectivas instruções SAVE, ao serem carregados da fita para o microcomputador os programas entrarão "rodando", tornando-se auto-explicativos através da apresentação ou do "menu", automaticamente introduzidos por eles próprios.

Recomendamos ao programador principiante interessado que estude detidamente cada programa, etapa por etapa, a fim de colher nos mesmos os exemplos de comandos e instruções que possam ajudá-lo em suas próprias programações.

GRAFICO COMPARATIVO DE VENDAS

```
GRAFICO COMPARATIVO VENDAS-83/84
       ▒ 1984 - ▮ 1983
                        VENDAS -
                        (MILHOES
                        DE CR$):

                        J   718.3
                        F   834.8
                        M   811.7
                        A  1055.8
                        M  1188.5
                        J  1377.8
                        J  1555.8
                        A  1755.8
                        S  1977.3
                        O  2475.8
                        N  2855.8
                        D  3277.3

                        TOTAL $:
                         20004.9
======== MESES ======== ========
```

```
   1 REM GRAFICO COMPARATIVO DE
VENDAS
   5 CLS
  10 LET T=0
  20 LET S=0
  30 LET M$="JFMAMJJASOND"
  40 DIM V(12)
  50 DIM W(12)
  60 FOR C=1 TO 12
  70 IF C=5 OR C=10 THEN CLS
  80 PRINT ,,"VENDAS DO MES ";C;
":";
  90 INPUT V(C
 100 PRINT V(C
 110 LET T=T+V(C)
 120 LET X=INT (T/100)
 130 PRINT ,,"ANO 1983  MES ";C
":";
 140 INPUT W(C
 150 PRINT W(C
 160 LET S=S+W(C)
 170 NEXT C
 180 CLS
 190 LET E=(PEEK 16404+256*PEEK
16405)-(PEEK 16400+256*PEEK 1640
1)
 200 IF E<184 THEN GOTO 590
 210 PRINT "GRAFICO COMPARATIVO
VENDAS-83/84"
 220 PRINT TAB 7;"▒ 1984 - ▮ 198
3"
 230 PRINT AT 21,0;"======== MES
ES ========"
 240 LET K=0
 250 FOR I=0 TO 22 STEP 2
 260 LET K=K+1
 270 FOR G=20 TO 20-INT (V(K)/X)
STEP -1
 280 PRINT AT G,I;"▒"
 290 NEXT G
 300 PRINT AT G,I;M$(K)
 310 NEXT I
 320 LET K=0
 330 FOR I=1 TO 23 STEP 2
 340 LET K=K+1
 350 FOR H=20 TO 20-INT (W(K)/X)
STEP -1
 360 PRINT AT H,I;"▮"
 370 NEXT H
 380 NEXT I
 390 PRINT AT 2,24;"VENDAS -";AT
3,24;"(MILHOES";AT 4,24;"DE CR$
):"
 400 PRINT
 410 LET T=0
 420 FOR I=1 TO 12
 400 PRINT TAB 24;M$(I);TAB 32-L
EN STR$ V(I);V(I)
 440 LET T=T+V(I)
 450 NEXT I
 460 PRINT
 470 PRINT TAB 24;"TOTAL $:"
 480 PRINT TAB 32-LEN STR$ T;T
 490 PRINT TAB 24;"========"
 500 STOP
 510 SAVE "VENDAS"
 520 PRINT AT 10,0;"GRAFICO ANTE
RIOR OU NOVO? (A/N)"
 530 LET R$=INKEY$
 540 IF R$="N" THEN RUN
 560 IF R$="A" THEN GOTO 180
 580 IF R$<>"N" OR R$<>"A" THEN
GOTO 530
 590 PRINT AT 10,2 "NAO EXISTE G
RAFICO ANTERIOR."
 500 PRINT ,,"ACIONE O COMANDO R
UN E NEW LINE."
```

PROGRAMAS FINANCEIROS

A listagem apresentada a seguir reúne os cinco pequenos programas apresentados no capítulo PROGRAMAS FINANCEIROS e reflete a aplicação das recomendações feitas no capítulo MANIPULANDO DIVERSOS PROGRAMAS A UM SÓ TEMPO.

A apresentação final do agrupamento é feita com o fim de torná-lo prontamente acessível ao leitor praticante, caso o mesmo encontre alguma dificuldade ao fazê-lo conforme recomendado.

```
PROGRAMAS FINANCEIROS

A - APLICACAO  DE  CAPITAL

I - APLICACAO   PROGRAMADA

J - TAXA DE  FINANCIAMENTO

M - VALOR PRESTACAO MENSAL

P - CADERNETA DE  POUPANÇA
```

```
   5 CLS
  10 PRINT AT 5,5;"PROGRAMAS FIN
ANCEIROS"
  20 PRINT ,,,,
  30 PRINT TAB 3;"A - APLICACAO
 DE  CAPITAL"
  40 PRINT
  50 PRINT TAB 3;"I - APLICACAO
  PROGRAMADA"
  60 PRINT
  70 PRINT TAB 3;"J - TAXA DE  F
INANCIAMENTO"
  80 PRINT
  90 PRINT TAB 3;"M - VALOR PRES
TACAO MENSAL"
 100 PRINT
 110 PRINT TAB 3;"P - CADERNETA
DE  POUPANCA"
 120 LET E$=INKEY$
 130 IF E$="A" THEN GOTO 200
 140 IF E$="I" THEN GOTO 500
 150 IF E$="J" THEN GOTO 800
 160 IF E$="M" THEN GOTO 1100
 170 IF E$="P" THEN GOTO 1400
 180 IF E$<>"A" OR E$<>"I" OR E$
<>"J" OR E$<>"M" OR E$<>"P" THEN
 GOTO 120
 200 CLS
 205 CLEAR
 210 PRINT "CAPITAL APLICADO?"
 220 INPUT CI
 230 PRINT ,,TAB 5;"CR$ ";CI
 240 PRINT ,,"TAXA MENSAL? - %
= ";
 250 INPUT I
 260 PRINT I
 270 PRINT ,,"PRAZO EM MESES? =
";
 280 INPUT T
 290 PRINT T
 300 LET CF=CI*(1+I/100)**T
 310 PRINT ,,"CAPITAL FINAL: CR$
";CF
 320 PRINT ,,"NOVO CALCULO? (S/N
)"
 330 INPUT R$
 340 IF R$="N" THEN GOTO 5
 350 CLS
 370 GOTO 210
 380 STOP
 500 CLS
 505 CLEAR
 510 LET CF=0
 520 PRINT "PARCELA MENSAL?"
 530 INPUT P
 540 PRINT ,,TAB 5;"CR$ ";P
 550 PRINT ,,"TAXA MENSAL? - %
= ";
 560 INPUT I
 570 PRINT I
 580 PRINT ,,"QUANTAS PARCELAS?
= ";
 590 INPUT T
 600 PRINT T
 610 FAST
 620 FOR E=T TO 1 STEP -1
 630 LET CA=P*(1+I/100)**E
 640 LET CF=CF+CA
 650 NEXT E
 660 SLOW
 670 PRINT ,,"CAPITAL FINAL ";T+
1;" º MES:";,TAB 5;"CR$ ";CF
```

```
 680 PRINT ,,"NOVO CALCULO? (S/N
)"
 690 INPUT R$
 700 IF R$="N" THEN GOTO 5
 720 CLS
 730 GOTO 510
 740 STOP
 800 CLS
 805 CLEAR
 810 PRINT "VALOR FINANCIADO?"
 820 INPUT V
 830 PRINT ,,TAB 5;"CR$ ";V
 840 PRINT ,,"VALOR DA PRESTACAO
?"
 850 INPUT P
 860 PRINT ,,TAB 5;"CR$ ";P
 870 PRINT ,,"Nº DE PRESTACOES?
= ";
 880 INPUT N
 890 PRINT N
 900 PRINT ,,"AGUARDE UNS INSTAN
TES..."
 910 LET I=((P*N/V)**(1/(N+1)))-
1
 920 LET J=(V*I)/(1-(1+I)**(-N))
 930 IF ABS (P-J)>01 THEN GOTO 9
60
 940 PRINT ,,"TAXA DE JUROS (%
): ";I*100
 950 GOTO 980
 960 LET I=I+((P-J)/P)*I
 970 GOTO 920
 980 PRINT ,,"NOVO CALCULO? (S/N
)"
 990 INPUT R$
1000 IF R$="N" THEN GOTO 5
1020 CLS
1030 GOTO 810
1040 STOP
1100 CLS
1105 CLEAR
1110 PRINT "VALOR A FINANCIAR?"
1120 INPUT V
1130 PRINT ,,TAB 5;"CR$ ";V
1140 PRINT ,,"TAXA DE JUROS? = "
;
1150 INPUT I
1160 PRINT I
1170 PRINT ,,"QUANTAS PARCELAS?
= ";
1180 INPUT N
1190 PRINT N
1200 LET P=V*((I/100)/(1-(1+I/10
0)**(-N)))
1210 PRINT ,,"VALOR DA PRESTACAO
:"
```



```
1220 PRINT ,,TAB 5;"CR$ ";P
1230 PRINT ,,"NOVO CALCULO? (S/N
)"
1240 INPUT R$
1250 IF R$="N" THEN GOTO 5
1270 CLS
1280 GOTO 1110
1290 STOP
1400 CLS
1405 CLEAR
1410 LET N=12
1420 DIM T(N)
1430 PRINT "VALOR APLICADO? - CR
$";
1440 INPUT V
1450 GOSUB 1730
1460 PRINT AT 0,32-LEN V$;V$
1470 PRINT
1480 FOR C=1 TO N
1490 IF C>=2 THEN GOSUB 1650
1500 IF C=7 THEN CLS
1510 PRINT "COR MON ";C;"º MES?:
";
1520 INPUT T(C)
1530 IF T(C)>20 THEN GOTO 1520
1540 PRINT T(C);
1550 PRINT "%"
1560 IF T(C)=0 THEN GOTO 1640
1570 PRINT "CAPITAL+COR+JUROS= "
;
1580 GOSUB 1620
1590 PRINT TAB 32-LEN V$;V$
1600 NEXT C
1610 IF C>N THEN GOTO 1640
1620 LET V=(V*T(C)/100+V)*1.005
1630 GOSUB 1730
1640 RETURN
1650 PRINT "SAQUE(-)/DEPOSITO?";
1660 INPUT K$
1670 LET S=VAL K$
1680 GOSUB 1780
1690 PRINT TAB 32-LEN S$;S$
1700 LET V=V+(S)
1710 GOSUB 1730
1720 RETURN
1730 LET V$=STR$ (V+.001)
1740 FOR I=1 TO LEN V$
1750 IF V$(I)<>"." THEN NEXT I
1760 LET V$=V$( TO I+2)+"0"
1770 RETURN
1780 LET S$=STR$ (S+.001)
1790 IF SGN S=-1 THEN LET S$=STR
$ (S-.001)
1800 FOR I=1 TO LEN S$
1810 IF S$(I)<>"." THEN NEXT I
1820 LET S$=S$( TO I+2)+"0"
```

```
1830 RETURN
1840 PRINT
1850 PRINT TAB 5 "VALOR FINAL: O
R$";TAB 32-LEN V$;V$
1860 PRINT ,,"NOVOS CALCULOS? (S
/N)"
1870 INPUT R$
1880 IF R$="N" THEN GOTO 5
1900 CLS
1910 GOTO 1410
1920 SAVE "POUPANCA"
1930 RUN
```


CADASTRO DE LIVROS

```
CADASTRO DE LIVROS JUNEL

I - INICIA  O CADASTRO

C - CONTINUA  CADASTRO

M - MODIFICA  REGISTRO

E - ELIMINA   REGISTRO

G - GRAVA OS REGISTROS

R - RELACIONA CADASTRO

T - BUSCA  PELO TITULO

A - BUSCA  PELO  AUTOR
```

```
 O CADASTRO DE LIVROS JUNEL PER-
MITE REGISTRAR  ATE' 200 TITULOS
E RESPECTIVOS AUTORES COM ATE 27
CARACTERES CADA. E' AUTOEXPLICA-
TIVO E AUTONOMO. SEU MENU INDICA
TODOS OS SEUS RECURSOS.

DUAS OBSERVACOES APENAS: NO CASO
DE <BREAK> ACIDENTAL DIGITE <GO-
TO 1300> E TECLE <NEW LINE> PARA
NAO PERDER OS REGISTROS JA' FEI-
TOS, E QUANDO APARECER, NO CANTO
INF. ESQ. DA TELA, O CODIGO 5/N'
...(FALTA DE ESPACO NA TELA) TE-
CLE <CONT> E <NEW LINE>.

DIGITE QUALQUER TECLA P/COMECAR
```

```
   0 REM CADASTRO DE LIVROS.  DO
LIVRO  "TK 85 - DOMINIO RAPIDO",
DE AUTORIA DE NELSON CASARI, SP.
  10 REM REPRODUCAO PROIBIDA. TO
DOS DIREITOS RESERVADOS POR LEI.
  20 CLS
  30 PRINT ,,,," O CADASTRO DE L
IVROS JUNEL PER-MITE REGISTRAR
ATE" 200 TITULOS"
  40 PRINT "E RESPECTIVOS AUTORE
S COM ATE 27CARACTERES CADA. E"
AUTOEXPLICA-TIVO E AUTONOMO. SEU
 MENU INDICATODOS OS SEUS RECURS
OS."
  50 PRINT ,,,,
  60 PRINT "DUAS OBSERVACOES APE
NAS: NO CASODE (BREAK) ACIDENTAL
 DIGITE <GO-TO 1300> E TECLE <NE
W LINE> PARANAO PERDER OS REGIST
ROS JA" FEI-"
  70 PRINT "TOS. E QUANDO APAREC
ER, NO CANTOINF. ESQ. DA TELA, O
 CODIGO 5/N",..(FALTA DE ESPACO
NA TELA) TE-CLE <CONT> E <NEW LI
NE>."
  80 PRINT ,,,,"DIGITE QUALQUER
TECLA P/COMECAR"
  90 PAUSE 4E4
 100 GOTO VAL "01300"
 110 LET N=200
 120 DIM T$(N,27)
 130 DIM A$(N,27)
 140 DIM Q$(27)
 150 FOR Y=1 TO N
 160 CLS
 170 IF Y>=N THEN GOTO VAL "0036
0"
 180 PRINT TAB 2;"P/PARAR REGIST
ROS DIGITE ST"
 190 PRINT ,,"TITULO ";Y;""LIVRO
:"
 200 INPUT T$(Y)
 210 IF T$(Y, TO 2)="ST" THEN GO
TO VAL "01300"
 220 PRINT ,,T$(Y)
 230 PRINT ,,"AUTOR: ";
 240 INPUT A$(Y)
 250 PRINT A$(Y)
 270 PRINT TAB 3;"R = REGISTRA /
 C = CORRIGE"
 280 INPUT D$
 290 IF D$="R" THEN GOTO VAL "00
340"
 300 IF D$="C" THEN GOTO VAL "00
320"
```



```
 310 IF D$<>"R" OR D$<>"C" THEN
GOTO VAL "00280"
 320 PRINT ,,TAB 8;"DIGITE DE NO
VO"
 330 GOTO VAL "00190"
 340 CLS
 350 NEXT Y
 360 PRINT ,,"ARQUIVO ESGOTADO"
 370 PRINT ,,"DIGITE NEW LINE"
 380 INPUT M$
 390 GOTO VAL "01300"
 400 CLS
 410 PRINT "N" DO REGISTRO A MOD
IFICAR?"
 420 INPUT D
 430 PRINT "N" ";D;" - CONSTA:"
 440 PRINT ,,T$(D)
 450 PRINT A$(D)
 460 PRINT ,,"DIGITE NOVO TITULO
"
 470 INPUT T$(D)
 480 PRINT "N" ";D;" = ";T$(D)
 490 PRINT ,,"DIGITE NOME DO AUT
OR:".
 500 INPUT A$(D)
 510 PRINT A$(D)
 520 PRINT ,,"OK, ORDEM EXECUTAD
A"
 530 PRINT ,,"DIGITE NEW LINE"
 540 INPUT M$
 550 GOTO VAL "01300"
 560 CLS
 570 PRINT "QUAL O N" DO REG. A
ELIMINAR?"
 580 INPUT K
 590 PRINT "N" ";K;" - CONSTA:"
 600 PRINT ,,T$(K)
 610 PRINT A$(K)
 620 LET T$(K)=""
 630 LET A$(K)=""
 640 PRINT ,,"OK, REGISTRO ELIMI
NADO"
 650 PRINT ,,"DIGITE NEW LINE"
 660 INPUT M$
 670 GOTO VAL "01300"
 680 CLS
 690 PRINT ,,"LISTAR DO N"? ";
 700 INPUT O
 710 PRINT O
 720 PRINT ,,"ATE QUE N"? ";
 730 INPUT M
 740 IF M>N THEN LET M=N
 750 PRINT M
 760 PAUSE 100
 770 CLS
```

```
 780 FOR J=0 TO M-1
 790 IF CODE T$(J)=0 THEN NEXT J
 800 FAST
 810 PRINT "LIVRO Nº ";J;": ";T$(
J)
 820 PRINT "AUTOR: ";A$(J)
 830 PRINT
 840 NEXT J
 850 SLOW
 860 PRINT ,,"DIGITE NEW LINE"
 870 INPUT M$
 880 FAST
 890 CLS
 900 SLOW
 910 GOTO VAL "01300"
 920 CLS
 930 PRINT ,,"TITULO? ";
 940 INPUT O$
 950 PRINT O$
 960 LET S=LEN O$
 970 PRINT ,,"AGUARDE UM POUCO"
 980 PAUSE 100
 990 CLS
1000 FOR I=1 TO N-1
1010 FAST
1020 IF T$(I, TO S)<>O$ THEN GOT
O VAL "01050"
1030 PRINT ,,T$(I)
1040 PRINT A$(I),"(Nº ";I;")"
1050 NEXT I
1060 SLOW
1070 PRINT ,,"DIGITE NEW LINE"
1080 INPUT M$
1090 GOTO VAL "01300"
1100 CLS
1110 PRINT ,,"AUTOR? ";
1120 INPUT O$
1130 PRINT O$
1140 LET S=LEN O$
1150 PRINT ,,"AGUARDE UM POUCO"
1160 PAUSE 100
1170 CLS
1180 FOR I=1 TO N-1
1190 FAST
1200 IF A$(I, TO S)<>O$ THEN GOT
O VAL "01230"
1210 PRINT ,,A$(I)
1220 PRINT T$(I),"(Nº ";I;")"
1230 NEXT I
1240 SLOW
1250 PRINT ,,"DIGITE NEW LINE"
1260 INPUT M$
1270 GOTO VAL "01300"
1280 SAVE "CADASTRO"
1290 GOTO VAL "00010"
```



```
1300 CLS
1320 PRINT ,,TAB 4;"CADASTRO DE
 LIVROS JUNIOR"
1340 PRINT ,,,,TAB 5;"I - INICIA
   O CADASTRO"
1360 PRINT ,,TAE 5;"C - CONTINUA
   CADASTRO"
1380 PRINT ,,TAB 5;"M - MODIFICA
   REGISTRO"
1400 PRINT ,,TAB 5;"E - ELIMINA
   REGISTRO"
1420 PRINT ,,TAB 5;"G - GRAVA OS
  REGISTROS"
1440 PRINT ,,TAB 5;"R - RELACION
A CADASTRO"
1460 PRINT ,,TAB 5;"T - BUSCA  P
ELO TITULO"
1480 PRINT ,,TAB 5;"A - BUSCA  P
ELO  AUTOR"
1490 LET R$=INKEY$
1500 IF R$="I" THEN GOTO VAL "00
110"
1510 IF R$="C" THEN GOTO VAL "00
160"
1520 IF R$="M" THEN GOTO VAL "00
400"
1530 IF R$="E" THEN GOTO VAL "00
560"
1540 IF R$="G" THEN GOTO VAL "01
280"
1550 IF R$="R" THEN GOTO VAL "00
680"
1560 IF R$="T" THEN GOTO VAL "00
920"
1570 IF R$="A" THEN GOTO VAL "01
100"
1580 IF R$<>"A" OR R$<>"C" OR R$
<>"E" OR R$<>"G" OR R$<>"I" OR R
$<>"M" OR R$<>"R" OR R$<>"T" THE
N GOTO VAL "01490"
```

CONTROLE DE ESTOQUE

CONTROLE DE ESTOQUE [illegible]

```
I - INICIA OS REGISTROS

C - CONTINUA  REGISTROS

A - ALTERA QTDE E CUSTO

B - BUSCA ITEM P/ELIMI-
       NAR - SUBSTUTUIR

R - RELACIONA ITENS COM
        DADOS E  $ TOTAL

G - GRAVA EM FITA "K-7"
```

```
  10 LET G=0
  20 LET A=275
  30 DIM A$(A,27)
  40 DIM I$(27)
  50 DIM B(A)
  60 DIM C(A)
  70 DIM D(A)
  80 FOR E=1 TO A
  90 CLS
 100 PRINT "P/PARAR REGISTROS DI
GITE "".""""
 110 PRINT ,,"ITEM? ";"(";E;")";
" = "
 120 INPUT A$(E)
 130 IF A$(E, TO 1)="." THEN GOT
O VAL "01470"
 140 PRINT A$(E)
 150 PRINT "QTDE.? = ";
 160 INPUT B(E)
 170 PRINT B(E)
 180 PRINT "CUSTO? = ";
 190 INPUT C(E)
 200 REM GOSUB VAL "09000"
 210 PRINT C(E)
 220 LET D(E)=B(E)*C(E)
 230 PRINT "VALOR CR$ = ";D(E)
```



```
 240 PRINT ,,"R = REGISTRA / C =
CANCELA"
 250 INPUT C$
 260 IF C$="R" THEN GOTO VAL "00
300"
 270 IF C$="C" THEN PRINT ,,"DIG
ITE DE NOVO"
 280 IF C$="C" THEN GOTO VAL "00
110"
 290 IF C$<>"R" OR C$<>"C" THEN
GOTO VAL "00250"
 300 CLS
 310 NEXT E
 320 CLS
 330 PRINT "QUAL A DENOMINACAO
DO ITEM?"
 340 INPUT I$
 350 PRINT I$
 360 LET J=LEN I$
 370 PRINT "AGUARDE"
 380 PAUSE 100
 390 CLS
 400 FOR I=1 TO A-1
 410 FAST
 420 IF A$(I, TO J)=I$ THEN GOTO
VAL "00470"
 430 NEXT I
 440 PRINT "NAO ENCONTRADO"
 450 SLOW
 460 GOTO VAL "00610"
 470 SLOW
 480 GOSUB VAL "01420"
 490 PRINT I;TAB 5;A$(I)
 500 PRINT TAB 4;B(I);TAB 10;C(I
);TAB 19;;"CR$";TAB 32-LEN STR$
D(I);D(I)
 510 PRINT ,,"DIGITE NOVOS DADOS
: "
 520 PRINT ,,"QTDE.? = ";
 530 INPUT B(I)
 540 PRINT B(I)
 550 PRINT "CUSTO? = ";
 560 INPUT C(I)
 570 PRINT C(I)
 580 LET D(I)=B(I)*C(I)
 590 PRINT "VALOR CR$ ";D(I)
 600 PRINT "OK, ALTERACAO FEITA"
 610 PRINT ,,"DIGITE NEW LINE"
 620 INPUT E$
 630 GOTO VAL "01470"
 640 CLS
 650 PRINT "BUSCA PELO Nº (N) OU
DENOMINACAO           (D) DO ITEM
?"
 660 INPUT G$
```

```
 670 IF G$="N" THEN GOTO VAL "00
700"
 680 IF G$="D" THEN GOTO VAL "01
070"
 690 IF G$<>"N" OR G$<>"D" THEN
GOTO VAL "00660"
 700 PRINT ,,"QUAL O Nº DO ITEM?
";
 710 INPUT I
 720 PRINT I
 730 PRINT ,,"CONSTA NESSE ITEM:
"
 740 PRINT
 750 GOSUB VAL "01420"
 760 PRINT I;TAB 5;A$(I)
 770 PRINT TAB 4;B(I);TAB 10;C(I
);TAB 19;"CR$";TAB 32-LEN STR$ D
(I);D(I)
 780 PRINT ,,"ELIMINA (E), SUBST
ITUE (S)         OU MANTEM (M)?
";
 790 INPUT H$
 800 PRINT H$
 810 IF H$="E" THEN GOTO VAL "00
850"
 820 IF H$="S" THEN GOTO VAL "00
900"
 830 IF H$="M" THEN GOTO VAL "01
030"
 840 IF H$<>"E" OR H$<>"S" OR H$
<>"M" THEN GOTO VAL "00780"
 850 LET A$(I)=""
 860 LET B(I)=0
 870 LET C(I)=0
 880 LET D(I)=0
 890 GOTO VAL "01030"
 900 CLS
 910 PRINT ,,"DIGITE NOVOS DADOS
"
 920 PRINT ,,"DENOMINACAO?"
 930 INPUT A$(I)
 940 PRINT A$(I)
 950 PRINT "QTDE.? = ";
 960 INPUT B(I)
 970 PRINT B(I)
 980 PRINT "CUSTO? = ";
 990 INPUT C(I)
1000 PRINT C(I)
1010 LET D(I)=B(I)*C(I)
1020 PRINT "VALOR CR$ = ";D(I)
1030 PRINT ,,"OK ORDEM CUMPRIDA"
1040 PRINT ,,"DIGITE NEW LINE"
1050 INPUT E$
1060 GOTO VAL "01470"
1070 PRINT "QUAL A DENOMINACAO D
O ITEM?"
```



```
1080 INPUT I$
1090 PRINT I$
1100 LET J=LEN I$
1110 PRINT "AGUARDE"
1120 PAUSE 100
1130 CLS
1140 FOR I=1 TO A-1
1150 FAST
1160 IF A$(I, TO J)=I$  THEN GOT
O VAL "01220"
1170 NEXT I
1180 PRINT "NAO ENCONTRADO"
1190 PRINT
1200 SLOW
1210 GOTO VAL "01040"
1220 SLOW
1230 GOTO VAL "00750"
1240 CLS
1250 PRINT "AGUARDE"
1260 PAUSE 100
1270 CLS
1280 FAST
1290 GOSUB VAL "01420"
1300 FOR I=1 TO A-1
1310 IF CODE A$(I)=0 THEN NEXT I
1320 PRINT I;TAB 5;A$(I)
1330 PRINT TAB 4;B(I);TAB 10;C(I
);TAB 19;"CR$";TAB 32-LEN STR$ D
(I);D(I)
1340 LET G=G+D(I)
1350 NEXT I
1360 PRINT ,,"      VALOR TOTAL ";
TAB 19;"CR$";TAB 32-LEN STR$ G;G
1370 SLOW
1380 LET G=0
1390 PRINT ,,"DIGITE NEW LINE"
1400 INPUT E$
1410 GOTO VAL "01470"
1420 PRINT "ITEM DENOMINACAO"
1430 PRINT TAB 4;"QTDE";TAB 10;"
CUSTO UN";TAB 19;"VALOR DO BEM"
1440 PRINT
1450 RETURN
1460 SAVE "CONTROLE"
1470 CLS
1480 PRINT ,,,,"       CONTROLE DE
ESTOQUE [illegible]"
1490 PRINT ,,,,"      I - INICIA
OS REGISTROS"
1500 PRINT ,,"      C - CONTINUA
 REGISTROS"
1510 PRINT ,,"      A - ALTERA QT
DE E CUSTO"
1520 PRINT ,,"      B - BUSCA ITE
M P/ELIMI-              NAR -
SUBSTUTUIR"
```

```
1530 PRINT ,,"      R - RELACIONA
 ITENS COM             DADOS
E  $ TOTAL"
1540 PRINT ,,"      G - GRAVA EM
FITA ""K-7""
1550 LET M$=INKEY$
1560 IF M$="I" THEN GOTO VAL "00
010"
1570 IF M$="C" THEN GOTO VAL "00
090"
1580 IF M$="A" THEN GOTO VAL "00
320"
1590 IF M$="B" THEN GOTO VAL "00
640"
1600 IF M$="R" THEN GOTO VAL "01
240"
1610 IF M$="G" THEN GOTO VAL "01
460"
1620 IF M$<>"A" OR M$<>"B" OR M$
<>"C" OR M$<>"G" OR M$<>"I" OR M
$<>"R" THEN GOTO VAL "01550"
```


BATALHA AR E MAR

BATALHA AR E MAR

PRODUCOES JUNEL

-NAVES EXTRA-TERRENAS TENTAM INVADIR O PAIS PELO MAR.

-QUANDO UMA NAVE ET SE APROXIMA A SIRENE E O RADAR DA BASE NAVAL DAO O ALARME. UM NAVIO SAI PARA COMBATE-LA.

-A BASE CONTA COM 04 NAVIOS. SE OS MESMOS FOREM DESTRUIDOS OS ETS INVADEM A TERRA.

-USE JOYSTICK OU AS TECLAS 5 E 8 PARA MOVIMENTAR O NAVIO E 0 PARA DISPARAR MISSEIS.

-POR CADA NAVE INIMIGA ATINGIDA: + 50 PONTOS.

-DISPAROS ERRADOS: - 25 PTS.

-NAVE INIMIGA POUSADA = NAVIO PERDIDO, E - 50 PONTOS.

+500 PONTOS = VITORIA
-300 PONTOS = DERROTA

```
  10 PRINT AT 7,5;"BATALHA"
  20 PRINT AT 12,6;"MAR E MAR"
  30 PRINT AT 19,8;"PRODUCOES  JUNEL"
  40 PRINT AT 21,6;"DIREITOS  RESERVADOS"
  50 FOR J=1 TO 200
  60 NEXT J
  70 CLS
  80 PRINT AT 5,2;"-NAVES EXTRA-TERRENAS TENTAM      INVADIR O PAIS PELO MAR."
  90 PRINT AT 9,2;"-QUANDO UMA NAVE ET SE APRO-      XIMA A SIRENE E O RADAR  DA      BASE NAVAL DAO O ALARME. UM      NAVIO SAI PARA COMBATE-LA."
 100 PRINT AT 15,2;"-A BASE CONTA COM 04 NAVIOS.      SE OS MESMOS FOREM DESTRUI-      DOS OS ETS INVADEM A TERRA."
 110 FOR J=1 TO 500
 120 NEXT J
 130 CLS
 140 PRINT AT 6,2;"-USE JOYSTICK OU AS TECLAS 5      E 8 PARA MOVIMENTAR O NAVIO      E 0 PARA DISPARAR  MISSEIS."
 150 PRINT AT 10,2;"-POR CADA NAVE INIMIGA ATIN-      GIDA: + 50 PONTOS."
 160 PRINT AT 13,2;"-DISPAROS ERRADOS: - 25 PTS."
 170 PRINT AT 15,2;"-NAVE INIMIGA  POUSADA = NA-      VIO PERDIDO, E - 50 PONTOS."
 180 PRINT AT 18,5;"+500 PONTOS = VITORIA           -300 PONTOS = DERROTA"
 190 FOR J=1 TO 500
 200 NEXT J
 210 CLS
 220 POKE 16418,0
 230 LET N=4
 240 LET PO=0
 250 LET X=20
 260 LET Y=15
 270 IF PO>300 THEN LET D=7
 280 IF PO<300 THEN LET D=3
 290 LET S=INT (RND*30)
 300 IF S<2 THEN LET S=2
 310 IF S>28 THEN LET S=28
 320 PRINT AT 0,0;"                                "
 330 PRINT AT 21,0;"                                "
 340 PRINT AT 21,0;"▒";AT 20,0;"▒";AT 19,0;"▒";AT 18,0;"I"
 350 PRINT AT 21,30;"▒▒";AT 20,30;"▒▒";AT 19,30;"▒▒";AT 18,30;"▒▒";AT 17,30;"▒"
 360 PRINT AT 22,0;"PONTOS:";AT 23,9;"NAVIOS:"
 370 FOR Q=0 TO 10
 380 PRINT AT 16,30;"▒ ";AT 17,0;"▒"
 390 PRINT AT 16,30;" ▒";AT 17,0;"▒"
 400 PRINT AT X,Y;"▒▒"
 410 NEXT Q
 420 FOR U=D TO X
 430 PRINT AT 16,30;"▒";AT 17,0;"▒"
 440 PRINT AT U,S;"▒"
```

```
 450 PRINT AT U,5;"▀▄"
 460 PRINT AT U,5;" "
 470 PRINT AT X,Y;"▄█▄"
 480 PRINT AT X,Y;"   "
 490 PRINT AT U+1,5;
 500 IF PEEK (PEEK 16398+256*PEE
K 16399)=8 THEN GOSUB VAL "00960
"
 510 LET Y=Y-(INKEY$="5")+(INKEY
$="8")
 520 IF INKEY$="" THEN PRINT AT
X,Y;"▄█▄"
 530 IF Y<1 THEN LET Y=1
 540 IF Y>30 THEN LET Y=30
 550 IF INKEY$="0" THEN GOSUB VA
L "00720"
 560 NEXT U
 570 PRINT AT 22,8;PO;"    "
 580 PRINT AT 23,18;N;"   "
 590 IF N=0 THEN GOTO VAL "01090
"
 600 IF PO<-300 THEN GOTO VAL "0
1090"
 610 IF PO<600 THEN GOTO VAL "00
270"
 620 IF PO>=600 THEN PRINT AT 22
,18;"VOCE VENCEU..."
 630 FOR J=1 TO 50
 640 NEXT J
 650 PRINT AT 22,18;"
 "
 660 PRINT AT 10,4;"QUER JOGAR D
E NOVO? (S/N)"
 670 IF INKEY$="" THEN GOTO VAL
"00670"
 680 IF INKEY$="S" THEN GOTO VAL
 "00210"
 690 IF INKEY$="N" THEN CLS
 700 PRINT AT 10,2;"OBRIGADO, AT
E A PROXIMA VEZ..."
 710 STOP
 720 FOR T=19 TO U STEP -3
 730 PRINT AT X,Y;"▄█▄"
 740 PRINT AT T,Y;"I"
 750 PRINT AT T,Y;" "
 760 IF T=U AND Y=5 THEN PRINT A
T U,5-1;"BUM"
 770 IF T=U AND Y=5 THEN PRINT A
T U,5-1;"   "
 780 IF T=U AND Y=5 THEN GOTO VA
L "00820"
 790 IF T=U AND Y<>5 THEN GOSUB
VAL "00890"
 800 NEXT T
 810 RETURN
```



```
 820 PRINT AT 22,16;"ACERTOU UMA
NAVE"
 830 PRINT AT X,Y;"▄█▄"
 840 FOR J=1 TO 50
 850 NEXT J
 860 LET PO=PO+50
 870 PRINT AT 22,15;"
 "
 880 GOTO VAL "00570"
 890 PRINT AT 22,15;"VOCE ERROU
O ALVO"
 900 PRINT AT X,Y;"▄█▄"
 910 FOR J=1 TO 50
 920 NEXT J
 930 LET PO=PO-25
 940 PRINT AT 22,15;"
 "
 950 RETURN
 960 FOR E=0 TO 10
 970 PRINT AT X,5;"▀▄"
 980 PRINT AT X,Y;"BUM"
 990 PRINT AT X,Y;"▒▒▒"
1000 PRINT AT X,Y;"BUM"
1010 PRINT AT X,Y;"   "
1020 PRINT AT 22,17;"NAVIO DESTR
UIDO"
1030 NEXT E
1040 PRINT AT X,5;" "
1050 PRINT AT 22,17;"
 "
1060 LET PO=PO-50
1070 LET N=N-1
1080 RETURN
1090 PRINT AT 22,18;"VOCE PERDEU
..."
1100 FOR J=1 TO 50
1110 NEXT J
1120 PRINT AT 22,18;"
 "
1130 GOTO VAL "00560"
1140 SAVE "BATALHA"
1150 RUN
```

DICAS PRÁTICAS

PADRONIZAÇÃO MONETÁRIA

1 - Até cinco dígitos de entrada, com mais até três decimais ou sem decimais:

```
10 INPUT A
20 LET A$=STR$ (A+.001)
30 PRINT A$ ( TO LEN A$-1)
```

2 - Até oito dígitos de entrada, sendo cortados os decimais. Acima de oito dígitos ocorre arredondamento do oitavo, sendo os demais apresentados como zeros (0).

```
   10 INPUT A
   20 LET A$=STR$ A
   30 FOR C=1 TO LEN A$
   40 IF A$(C)  "." THEN NEXT C
   50 PRINT A$ ( TO C-1)
ou 50 PRINT A$ ( TO C-1)+",00"
```

3 - Até treze dígitos de entrada, com ou sem decimais:

```
   10 INPUT A
   20 LET A$=STR$ (A+.0001)
   30 FOR C=1 TO LEN A$
   40 IF A$(C)<>"." THEN NEXT C
   50 IF (C+1)>LEN A$ THEN LET A$=A$+".000"
   60 IF (C+2)>LEN A$ THEN LET A$=A$+"00"
   70 IF (C+3)>LEN A$ THEN LET A$=A$+"0"
   80 PRINT A$ ( TO C+3)
ou 80 PRINT A$ ( TO C+3)+",00"
```

Obs.: O TK 85 usa no máximo treze (13) espaços para escrever um número, mas fornece apenas os oito primeiros dígitos significativos como resultado de operações matemáticas. Números com catorze (14) ou mais dígitos são representados em notação científica, como nas calculadoras científicas de bolso. Experimente as seguintes operações:

```
PRINT 12345678*123456  (= 1524148000000)
PRINT 12345678*1234567 (= 1.5241567E+13)
```

E+13 no resultado da segunda operação indica que o resultado correto da operação será obtido multiplicando-se 1.5241567 por 10 e levado à décima terceira (13ª) potência, portanto resultando um número de 14 dígitos, que não pode ser impresso pelo microcomputador: 15241567000000.

ESCRITA VERTICAL

1 - De uma palavra com até 22 caracteres:

```
10 DIM A$(1,22)
20 INPUT A$(1)
30 FOR C=1 TO 22
40 PRINT AT C-1,0;A$(1,C)
50 NEXT C
```

Para imprimir em vídeo inverso, acrescente:

15 POKE 16390, 116 e tecle NEW LINE duas vezes após digitar a palavra.

2 - De um texto corrido ou diversos títulos ou palavras avulsas:

```
10 LET C=0
20 LET L=0
30 INPUT A$
40 FOR V=1 TO LEN A$
50 IF PEEK 16442=2 THEN GOSUB 150
60 IF C=32 THEN GOSUB 200
70 PRINT AT L,C;A$(V)
80 LET L=L+1
```

```
90 NEXT V
100 LET C=C+2
110 GOTO 20
150 LET C=C+2
160 LET L=0
170 RETURN
200 LET C=0
210 LET L=0
220 CLS
230 RETURN
```

CENTRALIZAÇÃO DE TITULOS

```
10 INPUT A$
20 LET A=LEN A$/2
30 PRINT TAB 16-A;A$
```

Ou, com vídeo inverso:

```
10 POKE 16390,116
20 INPUT A$
30 PRINT TAB 16-LEN A$/2;A$
```

Tecle NEW LINE duas vezes após digitar a palavra.

TITULO EM SCROLL

```
10 LET A$= "DICAS PRATICAS"
20 PRINT AT 19,8;A$
30 FOR C=1 TO 100
40 NEXT C
50 FOR C=1 TO 19
60 SCROLL
70 NEXT C
80 GOTO 10
```

DESTAQUE DE TITULOS


```
10 INPUT A$
20 FOR A=1 TO 5
30 PRINT AT 10,16-LEN A$/2;A$
40 FOR C=1 TO LEN A$
50 LET B$=CHR$ (CODE A$(C)+128)
60 PRINT AT 10,16-LEN A$/2+C-1;B$
70 NEXT C
80 NEXT A
```

INVERSÃO DE PALAVRAS OU NUMEROS

```
10 INPUT A$
20 PRINT A$
30 FOR C=LEN A$ TO 1 STEP -1
40 PRINT A$(C);
50 NEXT C
```

ENCOLUNAMENTO DE NUMEROS PELA DIREITA

```
10 INPUT A
20 PRINT TAB 32-LEN STR$ A;A
```

ALINHAMENTO DE PALAVRAS PELA DIREITA

```
10 INPUT A$
20 PRINT TAB 32-LEN A$;A$
```

RESERVANDO ESPAÇO EM LINHA REM

1 - Até 255 espaços:

```
1 REM ..          NEW LINE
POKE 16514,255    NEW LINE
POKE 16515,0      NEW LINE
RAND USR 8192     NEW LINE
NEW LINE
```

2 - 256 ou mais espaços:

```
256:    1 REM ..           NEW LINE
        POKE 16514,0       NEW LINE
        POKE 16515,1       NEW LINE
        RAND USR 8192      NEW LINE
        NEW LINE

300     1 REM ..           NEW LINE
        POKE 16514,44      NEW LINE
        POKE 16515,1       NEW LINE
        RAND USR 8192      NEW LINE
        NEW LINE

550     1 REM ..           NEW LINE
        POKE 16514,38      NEW LINE
        POKE 16515,2       NEW LINE
        RAND USR 8192      NEW LINE
        NEW LINE
```

Impresso na
press grafic
editora e gráfica ltda.
Rua Barra do Tibagi, 444 - Bom Retiro
Cep 01128 - Telefone: 221-8317

TK 85 - DOMÍNIO RÁPIDO

Nélson Casari

Este livro proporciona ao usuário de microcomputadores da linha *Sinclair,* em especial do TK 85 e do CP 200 S, uma aprendizagem agradável e segura.

Trata-se de um manual prático e de fácil consulta, constituído de informações, dados, tabelas, programas e "dicas" de acesso rápido, que permite o domínio adequado dos recursos do equipamento disponível.

Pretende-se colocar ao alcance do leitor os elementos que o auxiliem a dominar mais rapidamente seu aparelho, possibilitando-lhe ampla utilização em espaço de tempo mais curto possível.

APLICAÇÃO

Manual destinado aos usuários de microcomputadores da linha *Sinclair,* em especial do TK 85 e do CP 200 S. Complementa o Manual do Fabricante, permitindo a utilização mais intensiva dos recursos desses equipamentos. Recomendado para o desenvolvimento de cursos de treinamento.

publicação atlas